

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет _____ *електроніки* _____
(повна назва)

Кафедра _____ *мікроелектроніки* _____
(повна назва)

«До захисту допущено»

Завідувач кафедри

(підпис)

(ініціали, прізвище)

“ _____ ” _____ 20____ р.

**Дипломна робота
на здобуття ступеня бакалавра**

зі спеціальності _____ *153 «Мікро- та наносистемна техніка»* _____
(код і назва)

на тему _____ *Керований регулятор температури для дослідної нагрівальної установки*

Виконав: студент _____ *4* _____ курсу, групи _____ *ДП-62*

_____ *Павлюк Владислав Петрович* _____

(прізвище, ім'я, по батькові)

(підпис)

Керівник роботи _____ *ст. викл. Надкерничний Степан Петрович* _____

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Консультанти:

з нормоконтролю _____ *доц., к.ф.-м.н., с.н.с. Георгій Свечніков* _____

(назва розділу)

(посада, наук. ступінь, вчене звання, прізвище, ін.)

(підпис)

з інформаційних питань _____ *доц., к.т.н., Юрій Діденко* _____

(назва розділу)

(посада, наук. ступінь, вчене звання, прізвище, ін.)

(підпис)

Рецензент _____ *доц. каф. ЕПС, к. т. н., с.н.с., Олександр Терлецький .* _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень
з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет електроніки

(повна назва)

Кафедра мікроелектроніки

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 153 «Мікро- та наносистемна техніка»

(код і назва)

Спеціалізація «Мікроелектронні інформаційні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

(ініціали, прізвище)

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

групи ДП-62

Павлюк Владислав Петрович

(група, прізвище, ім'я, по батькові)

1. Тема роботи Керований регулятор температури для дослідної нагрівальної установки

керівник роботи Надкерничний Степан Петрович, старший викладач,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Термін подання студентом роботи 01 червня 2020 р.

3. Вихідні дані роботи Керований регулятор температури в діапазоні температур від 0°C до 600°C з точністю $\pm 2^\circ\text{C}$

4. Перелік питань, які мають бути розроблені (зміст роботи) 1) Розгляд типових схем електронних регуляторів температури. 2) Вивчення принципів роботи та особливостей конструкцій датчиків температури, 3) Ознайомлення з методами побудови пристроїв для вимірювання та регулювання температури на основі мікроконтролера. 4) Освоєння принципів роботи та особливостей програмування мікроконтролерного пристрою. 5) Розробка структурної схеми регулятора температури, проектування принципової схеми. 6) Розробка програмного забезпечення. 7) Вибір та розрахунок компонентів і проектування загальної конструкції. 8) Проведення монтажу та випробування роботи регулятора, внесення коректив у схему регулятора. 8) Обговорення результатів роботи, та написання дипломної роботи

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, креслеників тощо)_

1. Типові схеми регуляторів температури.

2. Схема регулятора температури на мікроконтролері.

3. Структура та алгоритм програми мікроконтролера.

4. Малюнок комутації керованого регулятора температури.

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 25 лютого 2020 р.

Календарний план виконання роботи

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Ознайомлення з типовими схемами електронних регуляторів температури.	06.03.20	
2.	Вивчення особливостей конструкцій датчиків температури різних типів	18.03.20	
3.	Розробка структурної схеми регулятора температури.	26.03.20	
4.	Вивчення особливостей роботи та принципів програмування мікроконтролера типу	20.04.20	
5.	Проектування принципової схеми регулятора температури. Вибір та розрахунок компонентів.	12.05.20	
6.	Розробка та виготовлення загальної конструкції регулятора.	15.05.20	
7.	Проведення лабораторних випробувань. Обговорення результатів випробувань	20.05.20	
8.	Написання дипломної роботи	03.06.19	

Студент

(підпис)

В. П. Павлюк

(ініціали, прізвище)

Керівник роботи

(підпис)

С. П. Надкерничний

(ініціали, прізвище)

РЕФЕРАТ

Дипломна робота обсягом 75 ст. містить: 40 іл., 2 табл., 1 дод., 5 джерел.

Мета роботи – розробка простого та ефективного керованого регулятора температури для дослідної нагрівальної установки лабораторного призначення на основі програмованого мікроконтролерного управління.

В даній дипломній роботі здійснено розробку простого та ефективного керованого регулятора температури для дослідної нагрівальної установки лабораторного призначення на основі мікроконтролерної апаратної обчислювальної платформи Arduino (Ардуіно). В розробці програмного забезпечення регулятора температури використано безкоштовне програмне забезпечення Arduino IDE.

У основній частині розглянуто типові схеми електронних регуляторів температури, проаналізовано та розглянуто різні види температурних датчиків, їх недоліки та переваги. Освітлено принципи роботи тих чи інших термодатчиків.

Проведено вибір основних компонентів керованого регулятора температури та організовано їх підключення до платформи Arduino Uno. Представлена функціональна схема регулятора температури, описані його основні функції. Описується структура програмного коду в середовищі розробки ArduinoIDE.

Представлено рекомендації щодо використання результатів розробки.

Ключові слова: регулятор температури, температурний датчик, нагрівальна установка, термопара, мікроконтролер, Arduino.

ABSTRACT

Thesis of 75 pages contains: 40 illustrations, 2 tables, 1 appendix, 5 sources.

The purpose of the work is to develop a simple and effective controlled temperature controller for an experimental laboratory heating installation based on a programmable microcontroller control.

In this thesis, a simple and effective controlled temperature controller for experimental heating installation for laboratory purposes based on the microcontroller hardware computing platform Arduino (Arduino). Free Arduino IDE software was used to develop the temperature controller software.

In the main part the typical schemes of electronic temperature regulators are considered, different types of temperature sensors, their disadvantages and advantages are analyzed and considered. The principles of operation of certain temperature sensors are covered.

The main components of the controlled temperature controller are selected and their connection to the Arduino Uno platform is organized. The functional scheme of the temperature regulator is presented, its main functions are described. Describes the structure of program code in the ArduinoIDE development environment.

Recommendations for the use of development results are presented.

Key words: temperature regulator, temperature sensor, heating installation, thermocouple, microcontroller, Arduino.

Зміст

Перелік скорочень, умовних позначень, термінів	7
Вступ.....	8
Розділ 1. Види регуляторів температури	9
Розділ 2. Особливості структури та будови термодатчиків	11
2.1. Принципи роботи датчиків температури.....	11
2.2. Терморезистор	12
2.3. Напівпровідникові термометри опору	13
2.4. Термопари	15
2.4.1 Виникнення термо-ЕРС	19
Розділ 3. Схема електронного терморегулятора.....	24
3.1 Терморегулятор на ШІМ контролері	24
Розділ 4. Розробка мікроконтролерного регулятора температури. Вибір компонентів для регулятора температури.	28
4.1 Функціональна схема регулятора температури	28
4.2 Розробка принципової схема регулятора температури.....	29
4.3 Термопара з цифровим підсилювачем	33
4.4 Мікроконтролер регулятора температури	35
4.5 Модуль реле для управління нагрівачем	38
4.6 LCD дисплей та модуль ІІС I2c SPI	39
4.7 Платформа Arduino Uno R3	42
Розділ 5. Конструкція керованого регулятора температури для дослідної нагрівальної установки	46
5.1 Функціональна схема конструкції.....	46
5.2 Написання коду програми в Arduino IDE	48
5.3 Поверхневий монтаж компонентів.....	52
Загальні висновки.....	57
Перелік джерел та посилань.....	58
Додаток 1. Код програми.....	59

Перелік скорочень, умовних позначень, термінів

ЕРС - Електрорушійна сила;

АПЦ - аналого-цифровий перетворювач;

ШІМ - широтно-імпульсна модуляція.

ТКО – температурний коефіцієнт опору

SPI (*Serial Peripheral Interface*) — послідовний периферійний інтерфейс

Вступ

Регулятори температури для різноманітних нагрівальних установок (терморегулятори) є затребуваним обладнанням, без якого важко уявити роботу багатьох приладів побутового призначення та різного виробничого обладнання. Можна зустріти регулятори температури, які монтуються окремо або призначені для установки безпосередньо в корпусі різної техніки. Робота терморегулятора неможлива без точного і надійного вимірювання температури.

Існує велика кількість способів вимірювання температури різними датчиками, роботу та структуру яких буде розглянуто. На сьогоднішній день великого поширення набули різноманітні датчики температури, що працюють за різними принципами та використовуються у багатьох сферах для регуляції та контролювання температур. Точні виміри температури вимагаються для забезпечення максимальної тепловіддачі в перетворювачах сонячної енергії, виміри температури сприяють ощадливій витраті енергії в побуті і на виробництві. У заданих межах повинна підтримуватися температура для монтажу SMD-компонентів при виготовленні електронних пристроїв.

Датчики температури можуть призначатись і тільки для вимірювання температури. У випадку, якщо пристрій містить датчик температури у поєднанні з компонентами необхідними для відображення вимірюваної температури, то подібний пристрій може називатися термометром.

Метою роботи є розробка простого та ефективного керованого регулятора температури для дослідної нагрівальної установки лабораторного призначення.

Такі регулятори мають широке застосування у дослідній роботі, в тому числі для дослідження теплофізичних властивостей матеріалів та компонентів, для теплових випробувань розроблюваних електронних пристроїв та систем.

Розділ 1. Види регуляторів температури

Регулятори температури (або терморегулятори) є обладнанням, що застосовується як у побутових приладах так і в установках виробничого призначення. Існують різні види регуляторів температури, які відрізняються за принципом керування: регулятори з механічним керуванням; регулятори з електромеханічним керуванням; регулятори з електронним керуванням та ручною установкою режимів; програмовані регулятори з електронним керуванням за наперед установленою програмою режимів нагріву. Перший різновид передбачає можливість установки постійної температури, яка буде підтримуватися до тих пір, поки режим не зміниться вручну. Регулювання таких приладів здійснюється шляхом обертання коліщатка встановленого на корпусі (Рис.1). Дані прилади відрізняються досить високою похибкою, але завдяки помірній ціні користуються великим попитом.



Рис.1 - Механічні регулятори температури з термодатчиками.

Електронні терморегулятори оснащуються дисплеєм, який відображає поточну температуру (Рис.2). Такі пристрої коштують дорожче, але відрізняються

високою точністю. Крім цього вони нерідко дозволяють здійснювати програмування режимів нагріву.



Рис.2 - Приклад електронного терморегулятора з дисплеєм

Також бувають універсальні терморегулятори, які можуть приєднуватися практично до будь-якого обладнання. Такі пристрої позбавлені власного корпусу і являють собою електронну плату з невеликим дисплеєм, що відображає поточну температуру. Їх можна приєднувати практично де завгодно. Такі пристрої можуть контролювати системи нагрівання або охолодження.

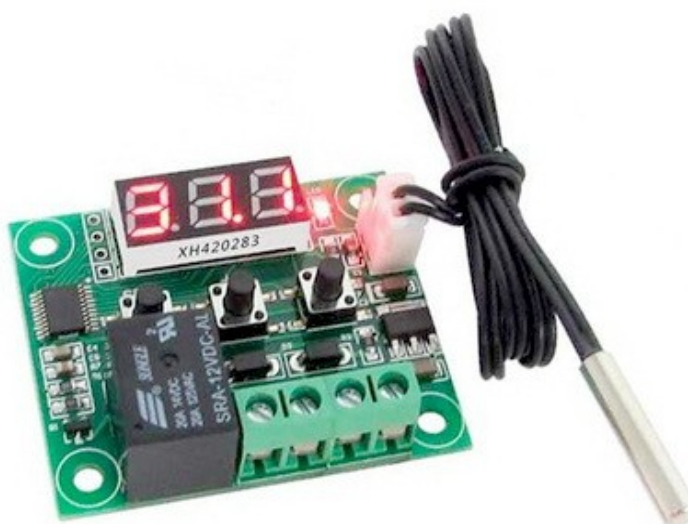


Рис.3 - Універсальний терморегулятор

Розділ 2. Особливості структури та будови термодатчиків

2.1. Принципи роботи датчиків температури

Температура макроскопічна характеристика макросистеми. Чисельно вона пропорційна середній кінетичній енергії молекул у речовині, яка приходить на одну ступінь свободи.

Термометр – така макроскопічна система, яка при віддачі чи поглинанні тепла у неї може змінюватися тільки один єдиний макроскопічний параметр. Такий параметр називається термічним параметром термометра.

В основі роботи термодатчиків використовується перетворення вимірюваних температур в електричні величини. Все це зумовлено такими причинами: електричну величину зручно передавати з великою швидкістю на значні відстані. Вимірювана температура може бути перетворена в цифровий код, що дозволяє нам досягнути великої швидкодії, чутливості та точності вимірювань.

Температура вимірюється контактним та безконтактним методами.

Для вимірювання температури контактним методом застосовують такі термометри:

- опору (використовують залежність електричного опору речовини від його температури);
- манометричні (використовують залежність тиску газів або насиченість парів рідини від температури);
- розширення (вимірюють температуру за тепловим розширенням рідин або твердих тіл);
- термоелектричні (основані на вимірюванні виникаючої термоелектрорушійної сили в місці контакту двох провідників).

Для вимірювання температури безконтактним методом використовують пірометри:

- кольорові (основані на вимірюванні відношення енергій, випромінюваних тілом в різних спектральних діапазонах);
- яркісні (вимірюють температуру за яркістю розжареного тіла в заданому вузькому діапазоні довжини хвиль);
- радіаційні (вимірюють температуру за тепловою дією сумарного випромінювання нагрітого тіла по всьому діапазону довжини хвиль).

За характером отримання інформації розрізняють пірометри для локального вимірювання температури в точці об'єкта та для аналізу температурних полів.

2.2. Терморезистор

Термін «термістор» або «терморезистор» утворився в результаті поєднання двох слів: «тепловий» і «резистор». Терморезистори відносяться до класу датчиків абсолютної температури, покази яких відповідають абсолютній температурній шкалі. Всі терморезистори поділяються на дві категорії:

- з від'ємним температурним коефіцієнтом опору;
- позитивним температурним коефіцієнтом опору.

Опір терморезистора залежить від температури. Він виготовлений з оксидів металів, пресованих у бісер, диск або циліндричну форму, а потім інкапсульований матеріалом, наприклад таким як скло. За допомогою терморезистора з від'ємним температурним коефіцієнтом опору при підвищенні температури опір зменшується. І навпаки, при зниженні температури опір збільшується. Цей тип терморезистора використовується найбільше. Терморезистор з позитивним температурним коефіцієнтом опору працює трохи інакше. При підвищенні температури опір збільшується, а при зниженні температури опір зменшується. Цей тип терморезистора зазвичай використовується як запобіжник. В діапазоні від 0°C до 650°C терморезистори з платини використовують як еталонні датчики вимірювань, похибка градуйованої характеристики платинових термоперетворювачів не перевищує 0,001°C.

Залежність опору платинових терморезисторів від температури можна визначити формулами:

$$R_t = R_0(1 + At + Bt^2) \text{ при } 0 < t < 650^\circ\text{C}, \quad (1.1)$$

$$R_t = R_0[1 + At + Bt^2 + Ct^3 (t-100)] \text{ при } -200^\circ\text{C} < t < 0^\circ\text{C}, \quad (1.2)$$

Платинові терморезистори володіють високою стабільністю і великою відтворюваністю характеристик. Недоліками є велика ціна та не лінійність перетворення.

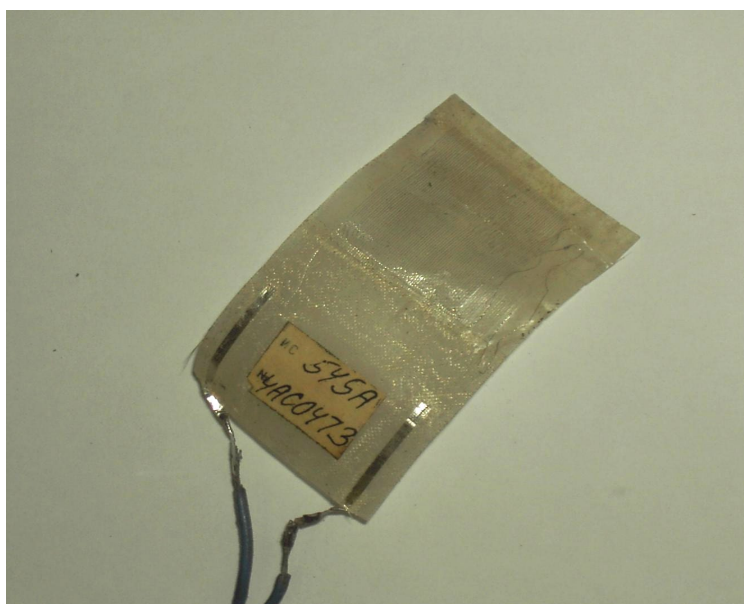


Рис.4 - Платиновий терморезистор плівкового типу

2.3. Напівпровідникові термометри опору

На відміну від металів багато напівпровідників і оксидів мають негативний температурний коефіцієнт опору. Залежність між величиною опору і температурою для таких часто є нелінійними.

Прикладом термометра з негативним ТКО є германієвий (Ge) термометр опору.

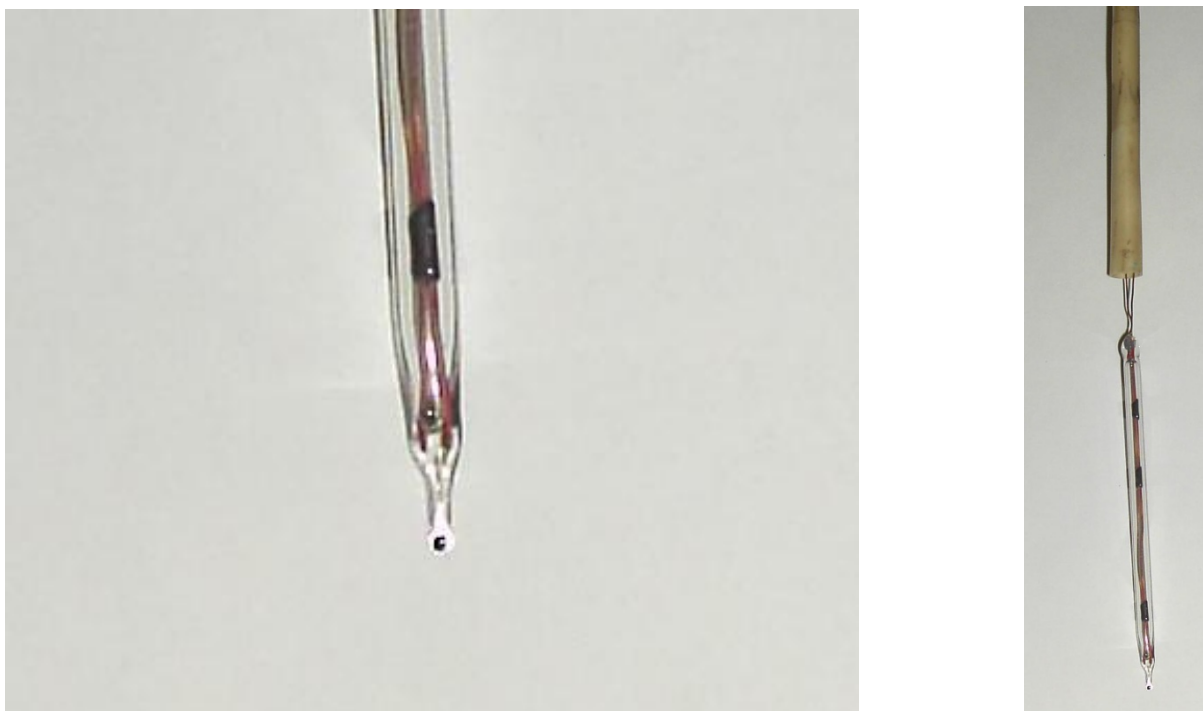


Рис.5 - Напівпровідниковий германієвий термометр опору

Напівпровідникові термоопори широко використовують для вимірів низьких температур (від 0,1K - 100K) в силу своєї високої чутливості. Представляють собою напівпровідникові пластини різних форм з прикріпленими металевими контактами, що зазвичай поміщають в захисні оболонки. У діапазонах температур від 4,2K до 13,8K використовують надточні германієві термоопори. При температурах вище 100K застосувати напівпровідникові термоопори не варто, бо з ростом температур змінюється і їх стабільність роботи.

Питома електрична провідність власного напівпровідника визначається класичним співвідношенням:

$$\sigma = n_o \mu_n q + p_o \mu_p q \quad (1.3)$$

Де n_o , p_o – рівноважна концентрація електронів та дірок;

μ_n , μ_p – рухливість електронів та дірок.

Для легованих напівпровідників концентрація основних носіїв завжди істотно більше, ніж концентрація неосновних носіїв.

Рухливість носіїв при нагріванні змінюється порівняно слабо (за степеневим законом, $\sim T^{3/2}$), а концентрація дуже сильно (експоненціально). Для власного напівпровідника концентрація носіїв заряду визначається наступною залежністю:

$$n_0 = p_0 = n_i = \sqrt{N_C \cdot N_V} \exp\left(-\frac{E_g}{2kT}\right) \quad (1.4)$$

Так як провідність напівпровідника безпосередньо залежить від числа вільних носіїв заряду, температурна залежність питомої провідності напівпровідника подібна температурній залежності концентрації основних носіїв.

Таким чином, опір напівпровідника буде залежати від температури за наступним законом:

$$R = N_o \cdot e^{\frac{\Delta E}{kT}} = N_o \cdot e^{\frac{B}{T}} \quad (1.5)$$

де T_1, T_2 – вихідна і кінцева температури робочого температурного діапазону;

R_1 і R_2 – опір терморезистора при температурах T_1 і T_2 .

2.4. Термопари

Термопари - це один із найчастіше використовуваних термодатчиків, які можуть використовувати для кількісного визначення температури. Їх широке використання для вимірювань температури пов'язане з їх низькою вартістю, надійністю та широким вимірюваним діапазоном температур. Як показано на рис. 6, термопара складається з двох несхожих за фізичними властивостями проводів 3 і 4. При різниці температур двох стиків, через ланцюг протікає невеликий струм, це вперше виявив Зеебек у 1823р. Цей струм пропорційний різниці температур між двома стиками, що часто називають "ефектом Зеебека". Причиною виникнення даного ефекту є різні значення роботи виходу електрона і різні концентрації вільних електронів у контактуючих металах. Термопари єдині температурні датчики, що дозволяють вимірювати температуру до 2300°C . Розроблено багато типів термопар, які класифікуються залежно від різних

комбінацій двох різних проводів (табл.1). Вибір певної термопари залежить від необхідного температурного діапазону, витрат, хімічної стійкості, сумісності матеріалів та механічної обмеження. До того ж, термопари мають велику лінійність та точність вимірювань.

Таблиця 1. Стандартизовані термопари

Тип	Температурний діапазон	Матеріал
T	Від -262°C до +850°C	Мідь / мідно-нікелевий сплав (константан)
J	Від -196°C до +700°C	Залізо / константан
E	Від -268°C до +800°C	Нікель-хромовий сплав (хромель) / константан
K	Від -250°C до +1100°C	Хром / нікель-алюмінієвий сплав (алюмель)
N	Від 0°C до +1250°C	Нікель-хром кремній / нікель-кремній–магнієвий сплав
B	Від +100°C до +850°C	Платина –30% родій/платина –60% родію
S	Від 0°C до +1500°C	Платина–10% родій/платина
R	Від 0°C до +1600°C	Платина – 13% родію/платини

Функція перетворення мідь-константанових термопар в діапазоні температур від -200°C до 300°C з погрешністю ± 2 мкВ описується формулою:

$$E = At^2 + Bt + C, \quad (1.7)$$

де A, B, C – сталі, які визначаються шляхом вимірювання термо-ЕРС при трьох відомих температурах;

t – температура робочого спаю, °C.

Схеми підключення термопари зображені на рис.6. У першому варіанті підключення (а) хоч точки і знаходяться за однакової температури, але з'єднані різнорідними провідниками, що в результаті може викликати некомпенсовану термо-ЕРС, що у свою чергу впливає на кінцевий результат вимірювань. При точних вимірюваннях у лабораторних умовах прилад підключається в розрив одного термоелектрода (електрода 4 на рис.6.б). Температуру вільного кінця термопари («опорного») при цьому підтримують постійною. Таке підключення компенсує непотрібну термо-ЕРС, що виникає у точках під'єднання термоелектрода з приладом. Підключення термопари таким способом називається диференціальним.

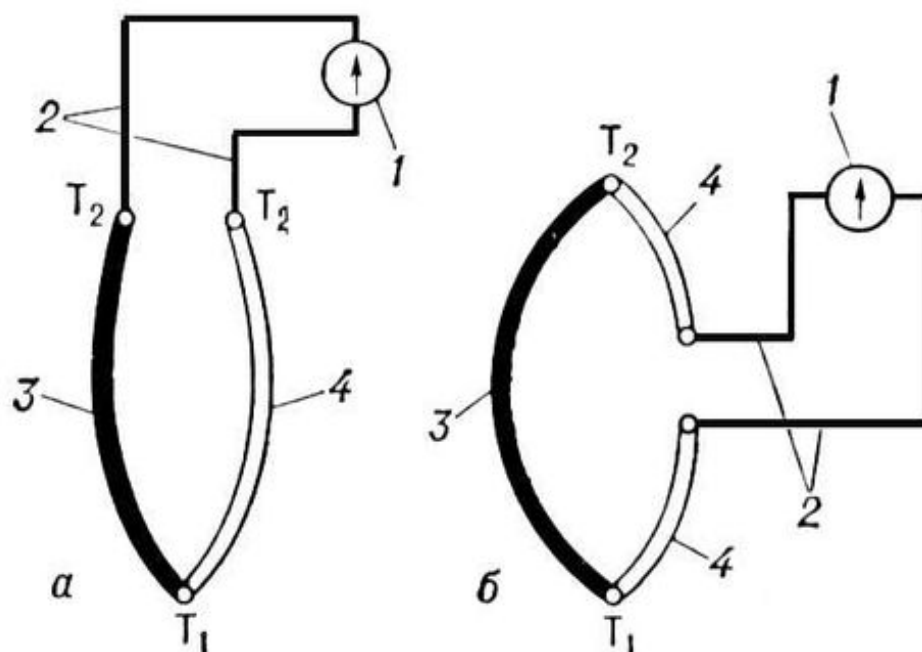


Рис.6 - Способи підключення термопари

Для розрахунку сумарної термо-ЕРС в ланцюзі диференціальної термопари необхідно знати температуру одного з спаїв. Раніше холодні спаї опускалися в посудину з водою та потовченим льодом (рис.7) для підтримки температури в 0°C (звідси з'явилася назва «холодний спай»), але це дуже незручно, а для багатьох практичних схем і зовсім неможливо. Тому на практиці температура холодного спаю контролюється додатковим датчиком, наприклад терморезистором.

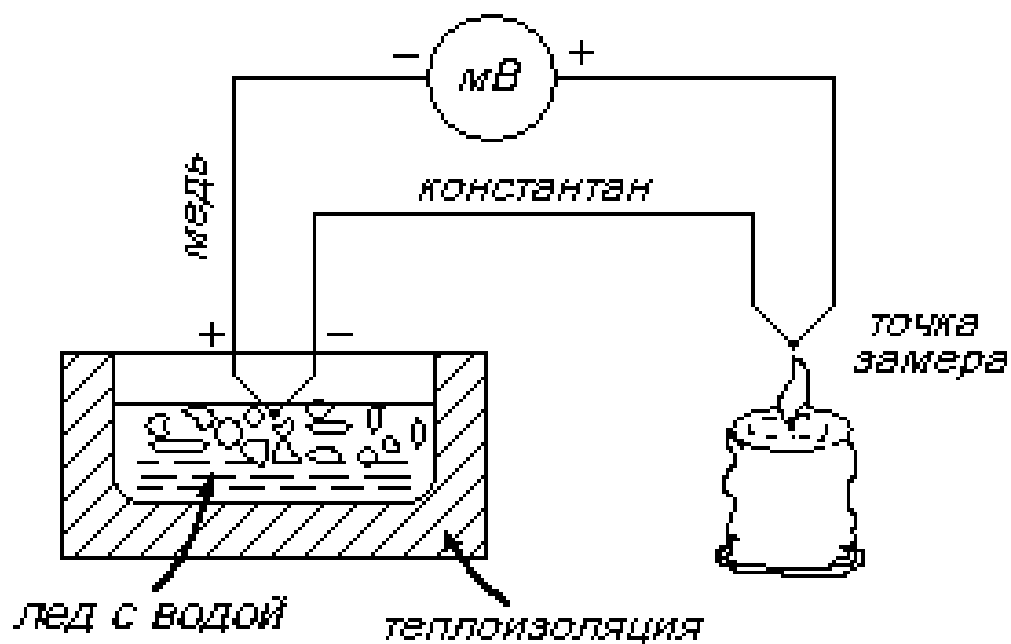


Рис.7 - Схема застосування диференційної термопари.

Залежність величини ЕРС термопари від температури, також як і опору металевого термометра опору від температури, нелінійні в широкому діапазоні температур, але на певній ділянці їх можна вважати лінійними і використовувати для розрахунків лінійні залежності.



Рис.8 - Проста конструкція термопари (а); вид контакту термопари (б)

2.4.1 Виникнення термо-ЕРС

На рис.9 наведені енергетичні діаграми двох металів з різною роботою виходу. На рис.9,а наведені енергетичні діаграми металів які не перебувають в контакті, а на рис.9,б – енергетичні діаграми контактуючих металів.

Метали обмінюються електронами, і так як енергія Фермі $E_F^B > E_F^A$ і робота виходу $\chi_A < \chi_B$, то переважними переходами будуть переходи електронів з металу В у метал А, в результаті чого в шарі сумарною товщиною d з'явиться надлишковий заряд, в металі А - негативний потенціал, а в металі В - позитивний. Між металами виникне контактна різниця потенціалів V_K , що дорівнює різниці робіт виходу з металів.

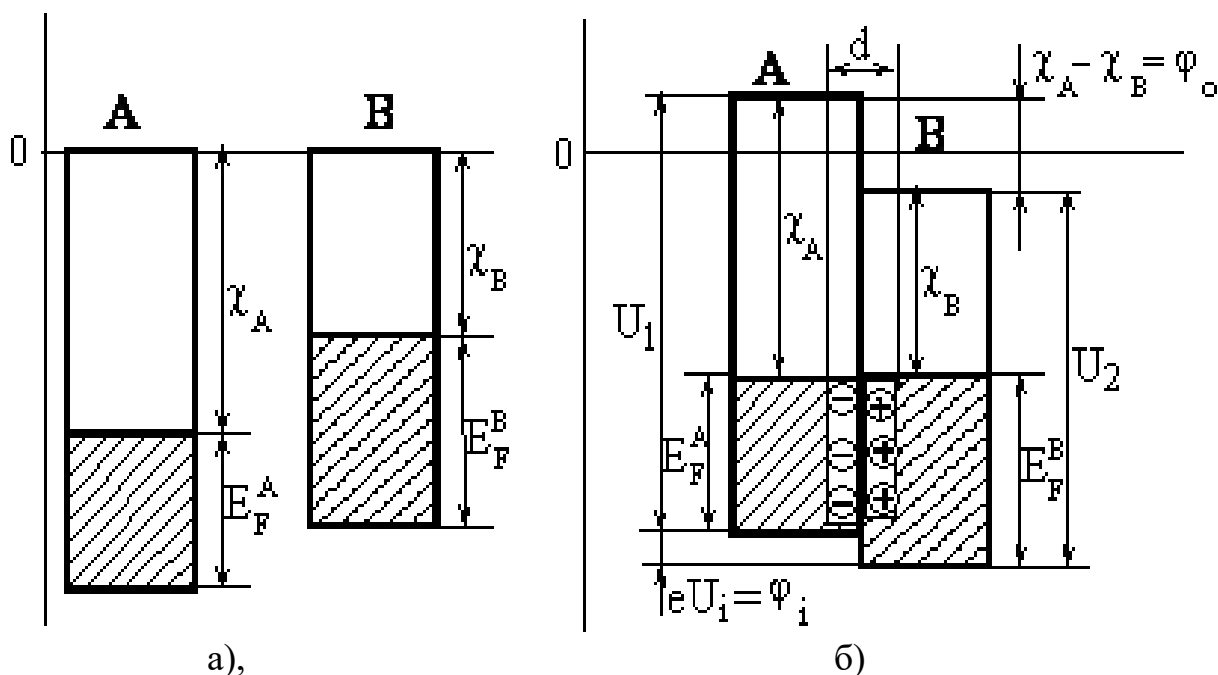


Рис.9 - Енергетична діаграма контакту двох металів

На рис.10. показано електричний ланцюг з двох різнорідних провідників, що мають контакти на протилежних кінцях.

При однаковій температурі контактів А і В різниці потенціалів V_K на кожному контакті, однакові за величиною і спрямовані назустріч один одному (рис.10, а). Потенціали врівноважуються і сумарна ЕРС в ланцюзі дорівнює нулю.

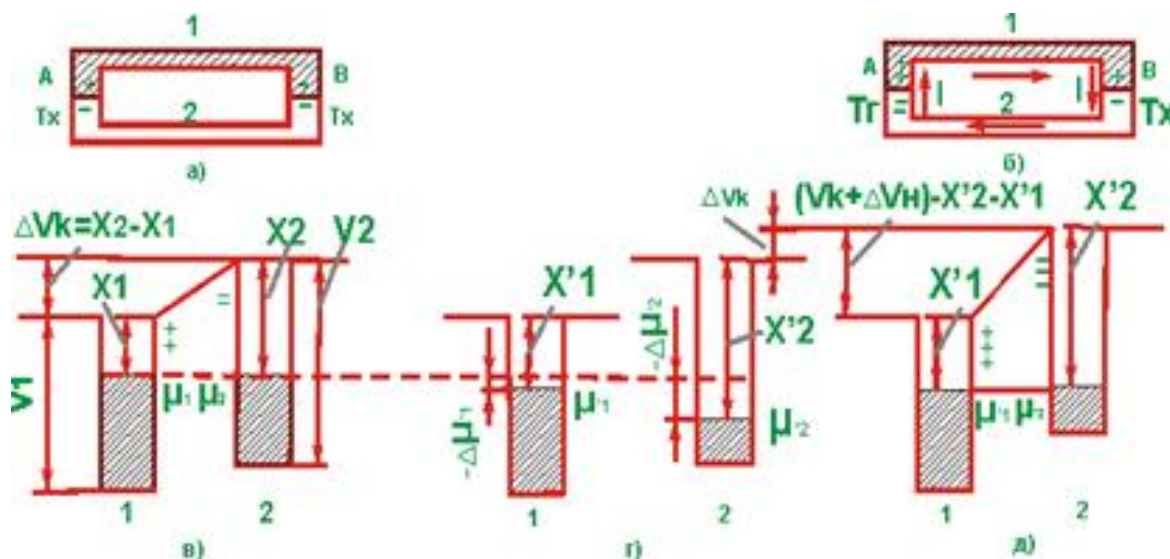


Рис.10 - Виникнення термоелектрорушійної сили в термопарі

Нагріємо контакт А, а В залишимо холодним. На контакті А енергетична діаграма зміниться (рис.9, б, г, д): контактна різниця потенціалів стане відмінною від тієї, яка існує на холодному контакті В, в ланцюзі виникне ЕРС.

ЕРС повного кола дорівнюватиме різниці ЕРС, що генеруються холодним і гарячим спаями, яка в свою чергу пропорційна різниці температур між спаями.

Якщо один спай двох металів (1 і 2) термопари має температуру T_X , а другий $T_Г$ (при цьому, наприклад, $T_Г > T_X$), то між кінцями термопари виникає термо-ЕРС E_{1-2} :

$$E_{1-2} = \alpha_{1-2} (T_Г - T_X) \quad (1.6)$$

Таблиця 2. Термо-ЕРС деяких термопар для різниці температур 100K (температура холодного спаю 0 °C).

Термопара	E_{1-2} , мВ
Залізо – константан	5,37
Мідь – константан	4,25
Ніхром – нікель	4,1
Ніхром – константан	6,21
Платина – платиновий	0,643
Залізо – мідь	1,05

2.5 Пірометри

Недоліками термометрів опору є необхідність введення датчика в досліджуване середовище. З цієї причини може змінюватися точність отримуваних даних і це призведе до швидшого виходження з ладу датчика та погіршення його характеристик. Ці недоліки відсутні у пірометрів – безконтактних датчиків, робота яких ґрунтується на фіксуванні випромінювань нагрітих тіл. Теплове випромінювання будь-якого характеризується кількістю енергії, випромінюваною тілом з одиниці поверхні в одиницю часу, і на одиницю діапазону довжини хвилі що припадає. Така характеристика представляє собою спектральну щільність і називається спектральною світністю. Закони температурного випромінювання точно визначені лише для абсолютно чорного тіла. Залежність спектральної світності абсолютно чорного тіла від температури і довжини хвилі виражається формулою:

$$R_{\alpha} = A\alpha^{-5}(e^{B/(\alpha T)} - 1)^{-1}, \quad (1.8)$$

де α – довжина хвилі;

T – абсолютна температура;

A і B – сталі величини.

Коли беруть будь-який об'єкт, температуру поверхні якого слід вимірювати пірометром, оптична система буде фіксувати енергію, випромінювану від об'єкта. Потім випромінювання направляється в детектор, цей елемент дуже чутливий до хвиль випромінювання. Вихідний сигнал детектора відповідає рівню температури об'єкта внаслідок випромінювання. Температура детектора, що аналізується за допомогою рівня випромінювання, прямо пропорційна температурі об'єкта.

Випромінювання, яке випромінюється з вимірюваного об'єкта, фактична температура якого повинна перевищувати абсолютну температуру (-273,15 градусів Цельсія). Це випромінювання називається інфрачервоним світлом в електромагнітному спектрі. Випромінювана енергія використовується для виявлення температури об'єкта і перетворюється в електричні сигнали за допомогою детектора.

Для виявлення температури різних об'єктів пірометри класифікуються на два типи:

- Оптичні пірометри;
- Інфрачервоні / радіаційні пірометри.

Оптичні пірометри це один із типів пірометрів, що використовуються для виявлення теплового випромінювання видимого спектру. Температура вимірюваних гарячих предметів буде залежати від видимого світла, яке вони випромінюють. Оптичні пірометри здатні забезпечити візуальне порівняння між каліброваним джерелом світла та досліджуваною поверхнею об'єкта. Наприклад, коли температура проводу та поверхні предмета однакова, то інтенсивність теплового випромінювання, спричинена завдяки проводу, зливається в поверхню цільового об'єкта і стає невидимою. Коли цей процес відбувається, струм, що проходить через провід, перетворюється на рівень температури.

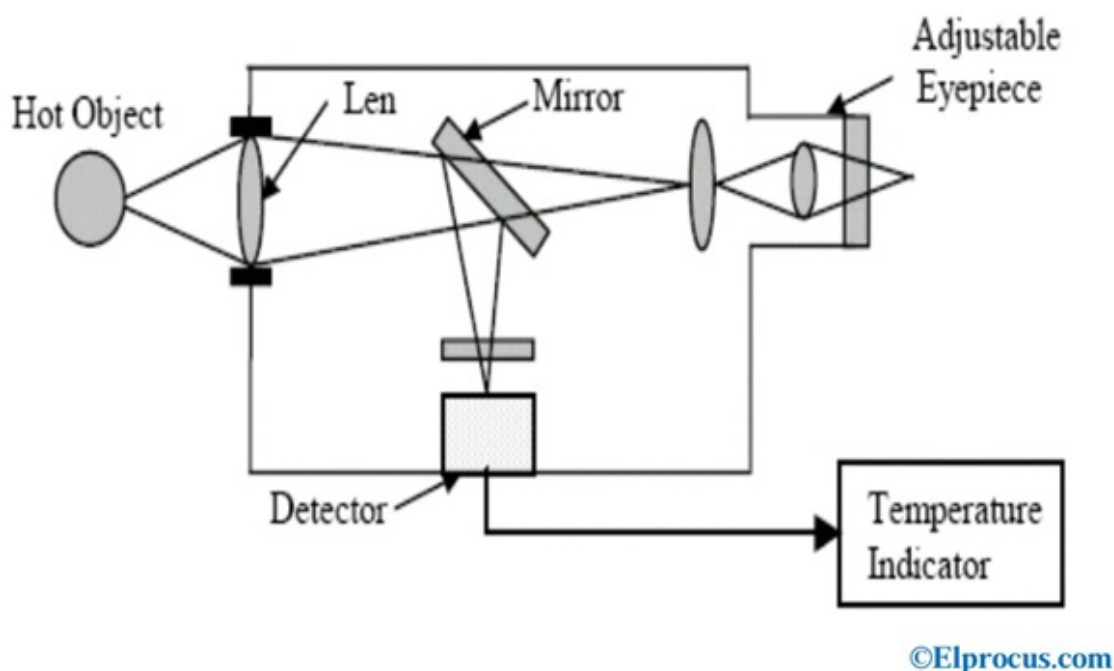


Рис.11 - Будова оптичного пірометра

Інфрачервоні або радіаційні пірометри призначені для виявлення теплового випромінювання в інфрачервоному спектрі випромінювання, що зазвичай знаходиться в межах від 2 до 14мкм. Він вимірює температуру об'єкта в

залежності від інтенсивності випромінювання. Це випромінювання може бути спрямоване на термопару для перетворення в електричні сигнали, оскільки термопара здатна генерувати більший струм, рівний випромінюваному теплу. Інфрачервоні пірометри складаються з піроелектричних матеріалів, таких як полівініліденфторид, тригліцину сульфат та танталат літію.

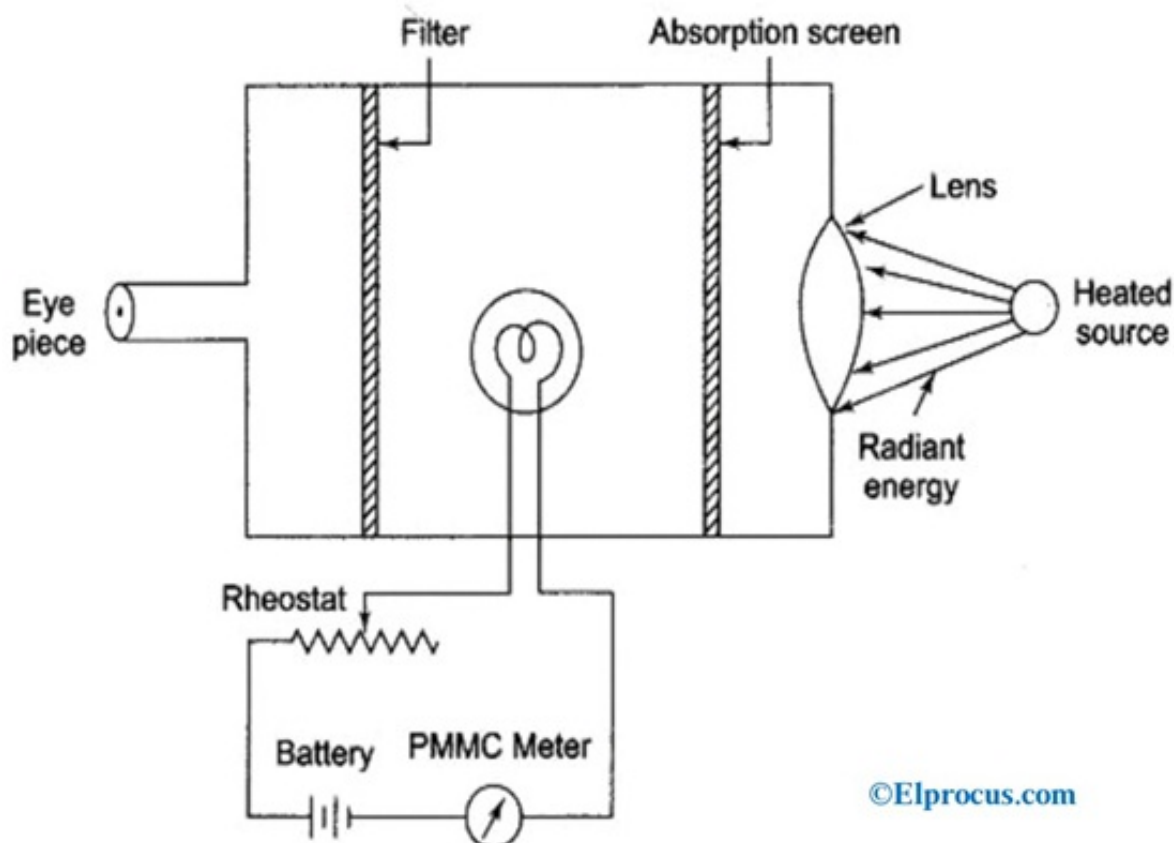


Рис.12 - Будова інфрачервоного пірометра

Перевагою пірометрів [2] є можливість вимірювання температури без безпосереднього контакту з досліджуваним об'єктом, швидкодія та вимірювання температур на різних відстанях навіть рухомих об'єктів. Недоліками можна вважати велику вартість та те, що на точність вимірювання може вплинути пил, дим та теплове випромінювання.

Розділ 3. Схема електронного терморегулятора

3.1 Терморегулятор на ШІМ контролері

Простий електронний терморегулятор для середніх температур до 150°C можна реалізувати на мікросхемі звичайного широтно-імпульсного контролера (ШІМ-контролера). Основою подібного терморегулятора служить мікросхема ШІМ-контролера, датчик температури і регулятор для установки температури стабілізації.

Для ШІМ-контролера можна застосувати відповідну мікросхему, наприклад типу KA3525A [4]. Застосування такого ШІМ-контролера обумовлене тим, що він може працювати із зовнішньою низькочастотною синхронізацією, має вбудоване джерело опорної напруги. Це джерело напруги використовується для живлення вихідного транзистора VT2 вузла, синхронізуючи його роботу з подвоєною частотою мережі. Датчиком температури такого пристрою (Рис.13) є прецизійна мікросхема ДК типу LM335A, що має хорошу лінійність характеристики в діапазоні температур від -40°C до $+100^{\circ}\text{C}$.

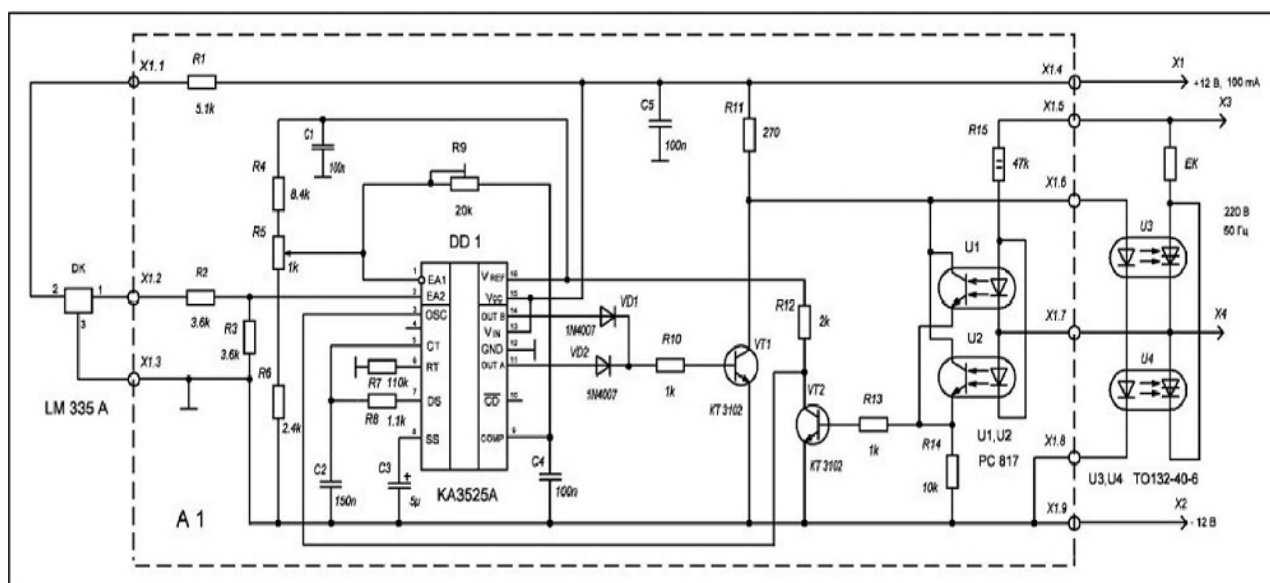


Рис.13 – Принципова схема терморегулятора на ШІМ контролері

Існуючі гальванічні ізольовані схеми формувачів коротких імпульсів при переході напруги через нуль мають велику кількість елементів, а значить, і відповідні габарити. Пропонована в [4] схема детектора переходу напруги живлення через нуль складається з двох транзисторних оптопар U1, U2 типу Pс817, світлодіоди які включені назустріч один одному, а фототранзистори - паралельно. При переході напруги через нуль, на колекторі вихідного транзистора VT2 формуються імпульси тривалістю 500мкс амплітудою 5В, які подаються на синхронізуючий вхід контролера КА3525А (вивід 3 DD1).

Напруга з виходу датчика температури DK через дільник напруги R1 R2 надходить на вивід 2 контролера DD1 (вхід підсилювача похибки). Підсилення підсилювача похибки регулюється підлаштовуванням резистором R9. З виходів 11 і 14 мікросхеми DD1 імпульси, тривалість яких пропорційна різниці заданої температури і температури робочого середовища, через суматор на діодах VD1, VD2 приходять на вхід інвертора VT1. З колектора цього транзистора імпульси надходять на послідовно з'єднані світлодіоди силових опотиристорів U3 і U4 типу TO132-40-6. Фототиристори цих елементів включені зустрічно-паралельно, тим самим забезпечують проходження змінного струму через нагрівач ЄК.

Установка необхідної температури здійснюється змінним резистором R5, а вибір діапазону робочих температур підбором резисторів R4 і R6, значення яких підбирають при налаштуванні терморегулятора. М'який пуск контролера здійснюється за допомогою конденсатора С3. Конденсатор С5 перешкоджає появі паразитних перешкод в ланцюзі харчування контролера.

Така побудова терморегулятора дозволяє звести до мінімуму перерегулювання температури в робочому обсязі з діапазоном температур від 35°C до 100°C. Точність підтримки температури становить $\pm 0,1^{\circ}\text{C}$.

Для живлення терморегулятора можна використовувати будь-який блок живлення напругою 12 В. Можна застосувати імпульсний блок живлення типу DSA-12 GFEU 120 120 з наступними вихідними параметрами:

- вихідна напруга = 12 В;
- струм навантаження до 0,3 А.

У пристрої можуть бути використані резистори потужністю 0,125 Вт, за винятком резистора R15, потужність якого повинна бути 2 Вт. Діоди VD1, VD2 типу 1N4007 можуть бути замінені КД521, КД522 або світлодіодами, за яскравістю яких можна судити про величину напруги, що надходить на навантаження. При виході на задану температуру світлодіоди перестають світитися. Термодатчик LM335A можна замінити LM35, при цьому необхідно провести перенастроювання контролера з підлаштування резистором R9. Довжина проводів, якими підключається термодатчик, не повинна перевищувати 1,5 м. Дві оптопари U1, U2 можна замінити однією подвійною типу PC827. Оптотиристори U3, U4 можна також замінити одним оптомодулем МТОТО 4 / 3-40-4.

Елементи пристрою А1 можуть бути розміщені на макетній платі розмірами 50х50 мм, яка за допомогою клемних колодок X1 і X2 підключається до інших елементів терморегулятора. Розміри корпусу терморегулятора визначаються величиною силових елементів U3, U4 і їх радіаторів.

Для настройки терморегулятора бажано виготовити стенд, що складається з дротяного трубчастого резистора типу С5-35 В або ПЗВ, потужністю 100 Вт і опором 680 Ом, який за допомогою стійок необхідно закріпити на ізоляційному підставі. При налаштуванні цей резистор використовується в якості навантаження. Для контролю температури необхідно застосувати ртутний термометр ТЛ-4, ціна ділення якого $+0,1^{\circ}\text{C}$.

Під час налаштування датчик температури і термометр знаходяться всередині навантажувального резистора. Регулювання пристрою зводиться до встановлення меж температурного діапазону в межах від 35°C до 100°C , а також градуювання шкали змінного резистора R4, який є задатчиком температури. Для установки температурного діапазону резистори R4 і R6 підбирають таким чином, щоб при верхньому положенні потенціометра R5 температура була максимальною, а при нижньому - мінімальною. Чутливість схеми встановлюють підлаштувальним резистором R9. Обертаючи ручку потенціометра R5, на його шкалі слід нанести значення температури відповідно до показань термометра. Якщо необхідно налаштувати терморегулятор в більш вузькому діапазоні

температур, то це можна здійснити зміною величин резисторів R4, R6. При цьому також треба заново відградувати шкалу задатчика температури.

При зазначених на схемі елементах, потужність термостата може досягати 8кВт. Для зміни вихідної потужності достатньо лише поміняти тип оптотиристорів U3, U4.

Пропонований пристрій може застосовуватися в побутових і промислових пристроях управління нагрівальними елементами для високоточної підтримки температури в робочій зоні. Пристрій має хороші динамічні характеристики. Крім того, в ньому забезпечена повна гальванічна розв'язка елементів терморегулятора від робочого середовища.

Розділ 4. Розробка мікроконтролерного регулятора температури.

Вибір компонентів для регулятора температури.

4.1 Функціональна схема регулятора температури

Існує велика кількість способів вимірювання температури різними датчиками. Великого поширення набули різноманітні датчики температури, що працюють за різними принципами та використовуються у багатьох сферах для регуляції та контролювання температур. Точні виміри температури вимагаються для забезпечення максимальної тепловіддачі в перетворювачах сонячної енергії, виміри температури сприяють ощадливій витраті енергії в побуті і на виробництві. У заданих межах повинна підтримуватися температура для монтажу SMD-компонентів при виготовленні електронних пристроїв, тому завданням роботи є розробка ефективного регулятора температури на основі сучасної елементарної бази.

Існує декілька видів регуляторів температури: механічні, електромеханічні, на базі спеціалізованих контролерів та програмовані мікроконтролерні регулятори температури. Будову керованого регулятора температури можна представити у вигляді узагальненої функціональної схеми (Рис.14).

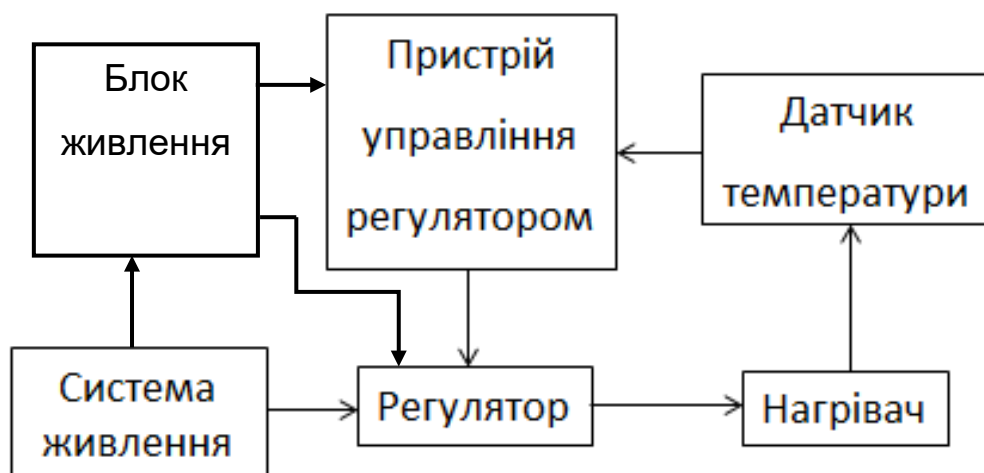


Рис.14 – Функціональна схема керованого регулятора температури.

Для дослідної нагрівальної установки найбільше підходить регулятор температури на базі програмованого мікроконтролера, тому що він має широкі можливості та універсальність при проектуванні, не потребує значних змін в конструкції при налаштуванні датчиків, можна ідентифікувати помилки за збої в процесі терморегуляції, можливе програмування під поставлені задачі.

Серед доступних датчиків температури, вибираємо термопару. Термопари мають достатньою чутливістю у широкому температурному діапазоні, стабільність градування, порівняно невелику інерційність, прості, надійні, можуть бути наведені в прямий контакт з вимірюваним об'єктом.

Регулювання температури нагрівача лабораторної установки найпростіше здійснювати за допомогою електромагнітного реле. Таке рішення є простим та надійним. На відміну від опторелейних та їм подібних рішень, електромагнітне реле не потребує додаткового пасивного або активного охолодження.

Нагрівальний елемент може бути представлений у вигляді резистивних нагрівачів (наприклад з ніхромових спіралей), або у вигляді інфрачервоних нагрівачів.

Система живлення нагрівача здебільшого забезпечується від промислової мережі 220 В, 50 Гц. Швидкість нагріву також може регулюватися комутуванням різних комбінацій з'єднання нагрівальних елементів (послідовне, паралельне, вибіркове).

4.2 Розробка принципової схема регулятора температури

Основним компонентом керованого регулятора температури є програмований мікроконтролер. На ринку представлено багато різноманітних мікроконтролерів. Одними з найпопулярніших є програмовані мікроконтролери типу Atmega. Для нашої розробки може бути застосований мікроконтролер типу ATmega328P, який представлено у двох різних корпусних рішеннях: PDIP28 та TQFP32.

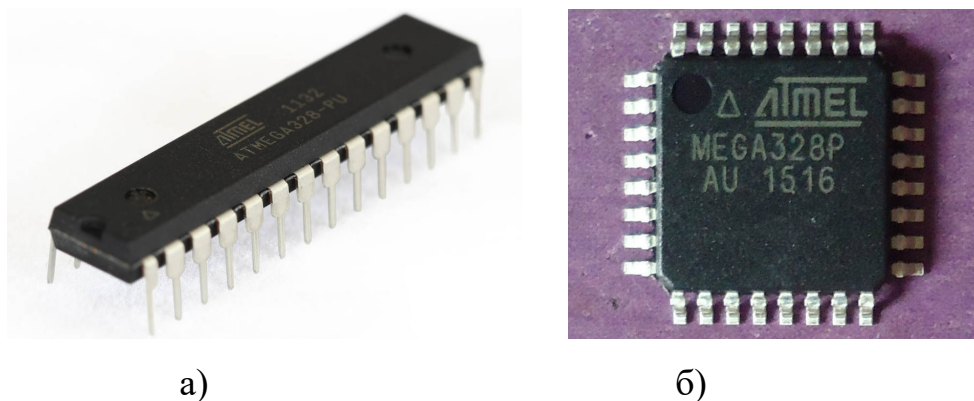


Рис.15 - Різні варіанти корпусного виконання мікроконтролера ATmega328P

а) - PDIP28, б) - TQFP32

Структурна схема мікроконтролера представлена рис.21. На рисунку структурної схеми представлені також основні входи і виходи мікроконтролера.

Роль датчика температури виконуватиме термопара К-типу, яка може забезпечувати діапазон вимірюваних температур від -250°C до $+1100^{\circ}\text{C}$. На рис.16 представлено під'єднання термопари та реле до мікроконтролера.

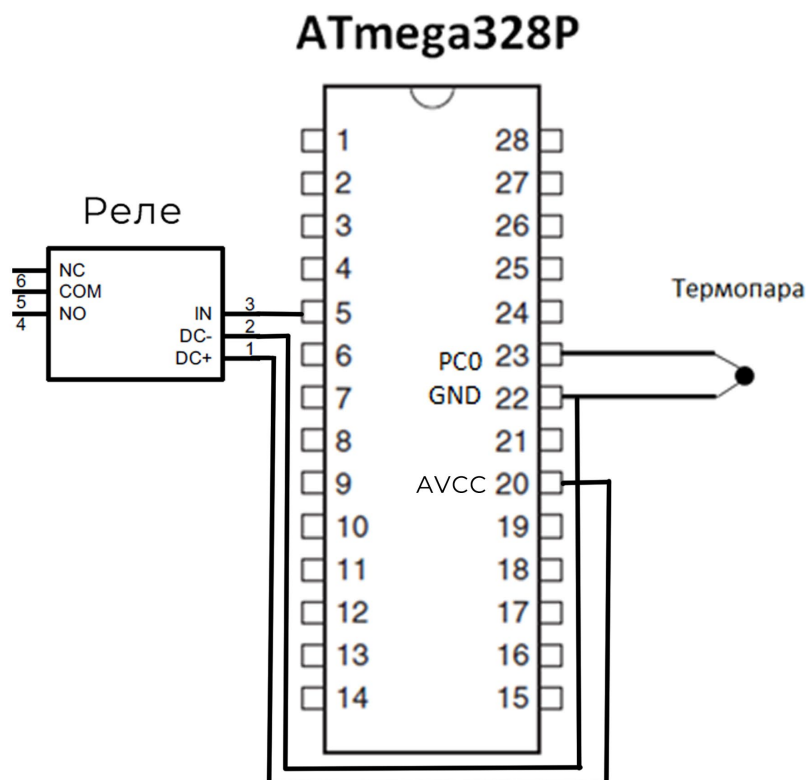


Рис.16 – Схематичне підключення реле та термопари до мікроконтролера.

Для програмування ATmega328P і реалізації необхідних функцій потрібен програматор. Щоб спростити собі задачу, було вирішено скористатися платформою Arduino Uno R3 на базі того ж ATmega328P, де присутні всі необхідні рішення. Це суттєво спростило процес програмування мікроконтролера.

Налаштування та моніторинг роботи регулятора температури відбуватиметься за допомогою LCD дисплею. Серед існуючих варіантів нам достатньо дисплею формату 16×2, для зручного розміщення інформації та навігації. Для зменшення часу розробки, розширення кількості портів введення або виводу для контролера Arduino Uno R3 та регулювання контрастністю поєднаємо LCD дисплей формату 16×2 з спеціальним I2C модулем.

Навігація здійснюватиметься трьома функціональними кнопками B1, B2, B3. Призначення кнопок описано в параграфі «5.2 Написання коду програми в Arduino IDE».

Для живлення Arduino Uno R3 та інших компонентів регулятора використаємо готове рішення у вигляді блоку живлення (адаптера AC/DC) на 9 В. Це значення напруги є оптимальним для нормального функціонування плати та підключених до неї модулів та датчиків.

На основі вибраних компонентів формується електрична схема усього пристрою для регулювання температури.

Загальна принципова схема програмно керованого регулятора температури для дослідної нагрівальної установки представлена на рисунку 17.

Детальний опис характеристик складових компонентів регулятора представлено у наступних окремих параграфах.

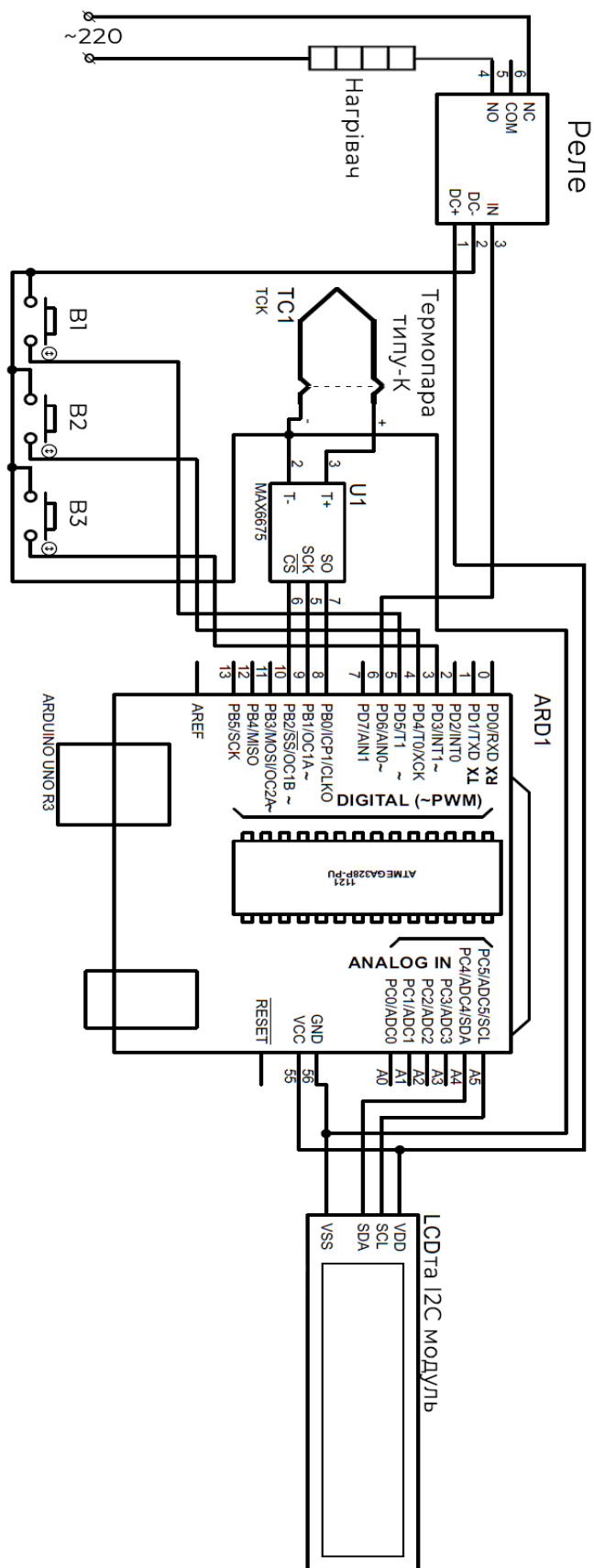


Рис.17 – Принципова схема регулятора температури для дослідної нагрівальної установки (створено в програмі «Proteus»).

4.3 Термопара з цифровим підсилювачем

У металевих термопар є недолік – відносно слабкий сигнал, який необхідно підсилити та необхідність компенсації холодного спаю. Також, у зв'язку з тим, що ми обрали мікроконтролерне рішення, бажано мати оцифрований сигнал термопари.

Можна вирішити ці проблеми вибравши вже готове модульне рішення - цифровий підсилювач на мікросхемі, який має інтегровану компенсацію холодного спаю, і на виході якого ми маємо цифровий сигнал.

Використаємо термопару К-типу в парі з цифровим підсилювачем MAX6675. Дана термопара (Рис.19) може бути використана для вимірювання температур від 0°C до 700°C. Застосований у мікросхемі MAX6675 (Рис.18) метод компенсації холодного спаю і цифровий метод відліку забезпечує високу точність вимірювання $\pm 0,25^{\circ}\text{C}$. До мікроконтролера модуль підсилювача можна підключити за шиною SPI (англ. *Serial Peripheral Interface* — послідовний периферійний інтерфейс).

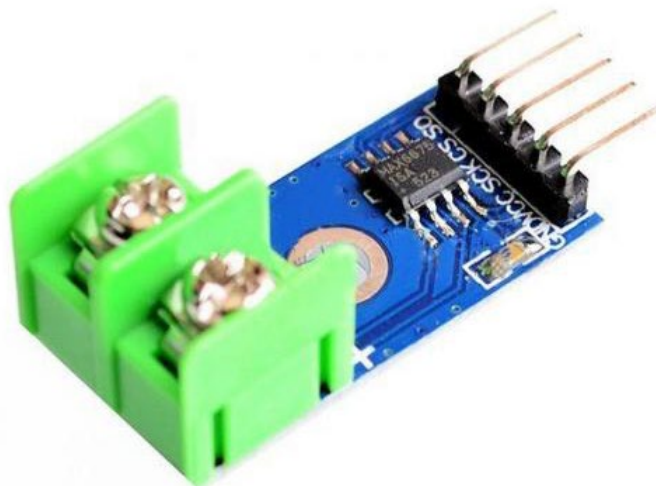


Рис.18 - Цифровий підсилювач сигналу термопари на мікросхемі MAX6675

Характеристики термопари К-типу в парі з MAX6675.

Напруга роботи: від 3.0 В до 5.5 В;

Мікросхема перетворювача: типу MAX6675;

Тип під'єднуваної термопари: К-тип;

Компенсація холодного спаю: інтегрована;

Інтерфейс: SPI;

Розрядність перетворювача: 12 біт;

Діапазон вимірюваної температури датчика: 0°C - 600°C ;

Температурний дозвіл: 0.25°C ;

Діапазон температури компенсації холодного спаю: від -20°C до $+80^{\circ}\text{C}$;

Контроль обриву термопари: інтегрований;

Довжина кабелю термопари 50 см;

Установча різьба датчика: M5x1 мм.



Рис.19 - Термопара типу-К (хромель-алюмелева)

Перевагами використання термопари типу хромель-алюмель є:

- широкий діапазон робочих температур, це самий високотемпературний з контактних датчиків.
- спай термопари може бути безпосередньо заземлений або наведено в прямий контакт з вимірюваним об'єктом.
- простота виготовлення, надійність і міцність конструкції та дешевизна.

4.4 Мікроконтролер регулятора температури

ATmega328P (Рис.20) це малопотужний 8-розрядний мікроконтролер, розроблений на основі AVR RISC архітектури. Мікроконтролер поставляється з прошитим завантажувачем Arduino Duemilanove (boot-loader), і готовий до роботи відразу після установки в вашу Arduino-сумісну плату, або в складі самостійно розробленого пристрою.

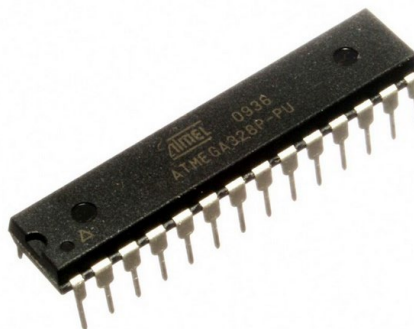


Рис.20 - Мікросхема мікроконтролера Atmega328P

Особливості роботи і структури (Рис.21) мікроконтролера ATmega328P.

- Високопродуктивний, малопотужний 8-бітний мікроконтролер AVR;
- Розширена архітектура RISC;
- 131 Потужна інструкція - виконання більшості одиничних циклів-годин;
- 32 x 8 робочих регістрів загального призначення;
- Повністю статична робота;
- Пропускна здатність до 20 MIPS на частоті 20 МГц;
- 2-цикльовий множник на мікросхемі;
- Елементи енергонезалежної пам'яті високої витривалості;
- 4/8/16 / 32К байт внутрішньосистемної самопрограмованої флеш-пам'яті;
- 256/512/512 / 1К байт EEPROM;
- Внутрішня SRAM 512 / 1К / 1К / 2К байтів;
- Цикли запису та стирання: 10000 флеш / 100 000 EEPROM;
- Збереження даних: 20 років при 85°C / 100 років при 25°C;

- Додатковий розділ завантажувального коду з незалежними блоками блокування;
- Внутрішньосистемне програмування за допомогою завантажувальної програми на мікросхемі;

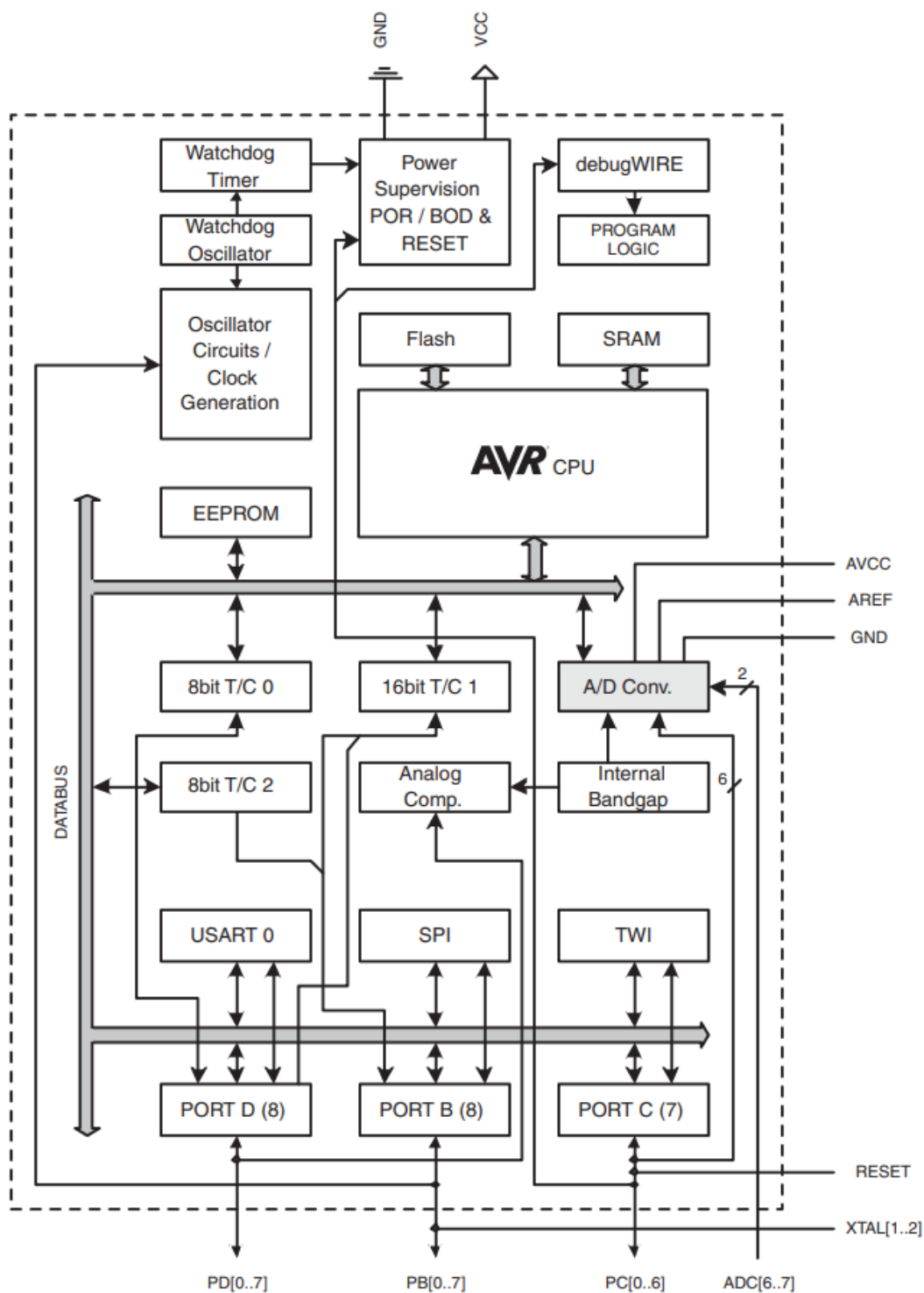


Рис.21 - Структурна схема мікроконтролера ATMEGA328P

- Справжня операція читання та запису;
- Два 8-бітні таймери / лічильники з роздільним попередньою програмою і режимом порівняння;
- Один 16-розрядний таймер / лічильник з окремим програмою дорожнього перегляду, порівняння режиму та захоплення;
- Лічильник реального часу з окремим осцилятором;
- Шість каналів ШІМ;
- 8-канальний 10-бітний АЦП в TQFP та QFN / MLF пакеті;
- Вимірювання температури;
- 6-канальний 10-бітний АЦП в PDIP-пакеті;
- Вимірювання температури;
- Програмований серійний USART;
- Послідовний інтерфейс головного / підлеглого SPI;
- 2-провідний послідовний інтерфейс (сумісний із Philips I2C);
- Програмований сторожовий таймер з окремим генератором на мікросхемі;
- Аналоговий компаратор на мікросхемі;
- Переривання та пробудження при зміні контактів;
- Скидання включення та програмоване виявлення коричневого відключення;
- Внутрішній калібрований осцилятор;
- Шість режимів сну: режим очікування, зменшення шуму АЦП, економія енергії, вимкнення живлення, очікування;
- розширений режим очікування;
- 23 програмовані лінії вводу / виводу;
- 28-контактний PDIP, 32-провідний TQFP, 28-контактний QFN / MLF та 32-контактний QFN / MLF;
- Робоча напруга 1,8 - 5,5В;
- 1,8 - 5,5В;
- Температурний діапазон: від -40°C до 85°C;

- Швидкість швидкодії: 0 - 4 МГц при 1,8 - 5,5 В, 0 - 10 МГц при 2,7 - 5,5В, 0 - 20 МГц при 4,5 - 5,5В;
- Споживання енергії на 1МГц, 1,8В, 25°C;
- Активний режим: 0,2 мА;
- Режим відключення живлення: 0,1 мкА;
- Режим економії енергії: 0,75 мкА (включаючи 32 кГц RTC).

4.5 Модуль реле для управління нагрівачем

Модуль реле, який має можливість змінювати полярність керуючого сигналу, а для його керування достатньо 5мА на керуючому сигналі. Встановлено струмообмежуючий резистор, що дозволяє забезпечити додатковий захист на вході і обмежити потік струму на оптроні гальванічної розв'язки.

Контакти модуля реле (Рис.22)

- DC(-, +): входи негативного та позитивного виходів джерела живлення;
- VREF: керування полярністю керуючого сигналу;
- CH1: керуючий сигнал.

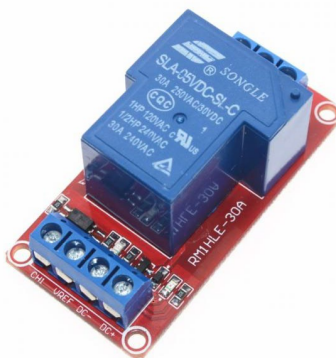


Рис.22 - Модуль реле

Також управляти реле можна подавши напругу на входи VREF та CH1. Значення поданої напруги можуть бути від 2,5В до 5В. Полярність при підключенні може бути будь-яка.

Характеристики реле.

- Робоча напруга котушки: 5В;

- Опір котушки: 150Ом;
- Струм споживання: 185мА;
- Номінальний струм котушки: 77мА;
- Максимальний комутований струм: 30А;
- Максимальна комутована напруга: 30В (постійна), 250В (змінна);
- Робоча температура: від -25°C до 70°C;
- Операційний час: 10мс;
- Час відпускання: 5мс;
- Розміри - 3.2 x 2.7 x 2.4 см;
- Вага: 5г.

Перевагами такого електромагнітного реле є те, що воно має малу вартість, комутація великого навантаження при невеликому розмірі пристрою, мала кількість випромінюваного тепла.

4.6 LCD дисплей та модуль ІС І2с SPI

LCD дисплей (Рис.23), з можливістю підключення до Arduino. Має два рядки, в кожному по 16 символів. Працює разом з бібліотекою LiquidCrystal.

.Характеристики LCD дисплея.

- Розміри 80мм x 36 мм;
- Температура роботи від 0°C до 50°C;
- Блакитна підсвітка;
- Білий колір символів;
- Розмір символів 4,35 x 2,95мм;
- Формат дисплею 16 x 2;
- Розміри точок 0,5мм x 0,5мм;
- інтерфейс HD44780;
- Видима область 64,5 x 13,8мм;
- Напруга живлення 5В.



Рис.23 - LCD дисплей

I2C інтерфейсний модуль (Рис.24) на мікросхемі PCF8574T розширює кількість портів введення або виводу для контролерів Arduino, мікроконтролерів STM8 і STM32 і мінікомп'ютерів Raspberry Pi і Orange Pi. Можна використовувати як інтерфейсну плату для підключення LCD 1602 і 2004, а також як самостійний пристрій. Щоб регулювати контрастність зображення є встановлений змінний резистор. Для нормальної роботи дисплея потрібно відрегулювати його контрастність. При використанні інтерфейсного модулю як розширювача портів введення або виводу слід враховувати те, що вихід P3 має інверсний вихід з відкритим колектором.

Характеристики інтерфейсного модуля I2C.

- Мікросхема інтерфейсу: PCF8574AT
- Інтерфейс: I2C;
- Діапазон адрес I2C:
 - PCF8574T - 0x20-0x27;
 - PCF8574AT - 0x38-0x3f.
- Кількість можливих підключень однотипних модулів: 8;
- Напруга живлення: 5 В;
- Розмір: 5.2см x 1.8см x 1.4см;



Рис.24 - I2C модуль для LCD дисплея.

Переваги використання інтерфейсного модуля I2C.

- Робота здійснюється двома лініями – лінією даних та синхронізації (SDA і SCL відповідно).
- Підключення великої кількості провідних приладів.
- Зменшення часу розробки.
- Управління пристроїв здійснюється єдиним мікроконтролером.
- Можливе число підключених мікросхем до однієї шині обмежується тільки граничною ємністю.
- Високий ступінь збереження даних через спеціальний фільтр, що нівелюють сплески, вбудований в схеми.
- Проста процедура діагностики виникаючих збоїв, швидке налагодження несправностей.
- Шина вже інтегрована в саму Arduino, тому не потрібно розробляти додатково шинний інтерфейс.

4.7 Платформа Arduino Uno R3

Плата Arduino Uno (Рис.25, Рис.26), це платформа контролера побудована на основі схеми мікроконтролера ATmega328. Платформа має 14 цифрових входів/ виходів і 6 з них можна використати як виходи ШІМ, 6 входів аналогових, генератор кварцовий який має частоту 16 МГц, має з'єднання через USB, роз'єм силовий, має ICSP та кнопку перезавантаження (Рис.25). Для програмування платформи її необхідно підключити до комп'ютера за допомогою USB кабелю. Живлення відбувається через USB або через адаптер AC/DC.

Характеристики:

- Мікроконтролер ATmega328P;
- Напруга роботи 5В;
- Напруга на вході від адаптер (рекомендується) 7-12В;
- Напруга на вході (межа) 6-20В;
- Цифрових контактів 14 (6 забезпечують вихід ШІМ);
- ШІМ-цифрові штифти вводу / виводу 6;
- Аналогові входні штифти 6;
- Постійний струм на контакт вводу/виводу 20 мА;
- Постійний струму для 3,3В Pin - 50 мА;
- Флеш пам'ять 32 КБ (ATmega328P), з яких 0,5 КБ для завантажувача;
- SRAM 2 КБ (ATmega328P);
- EEPROM 1 Кб (ATmega328P);
- Тактова частота 16 МГц;
- LED_BUILTIN 13;
- Довжина 68,6 мм;
- Ширина 53,4 мм;
- Вага 25 грам.

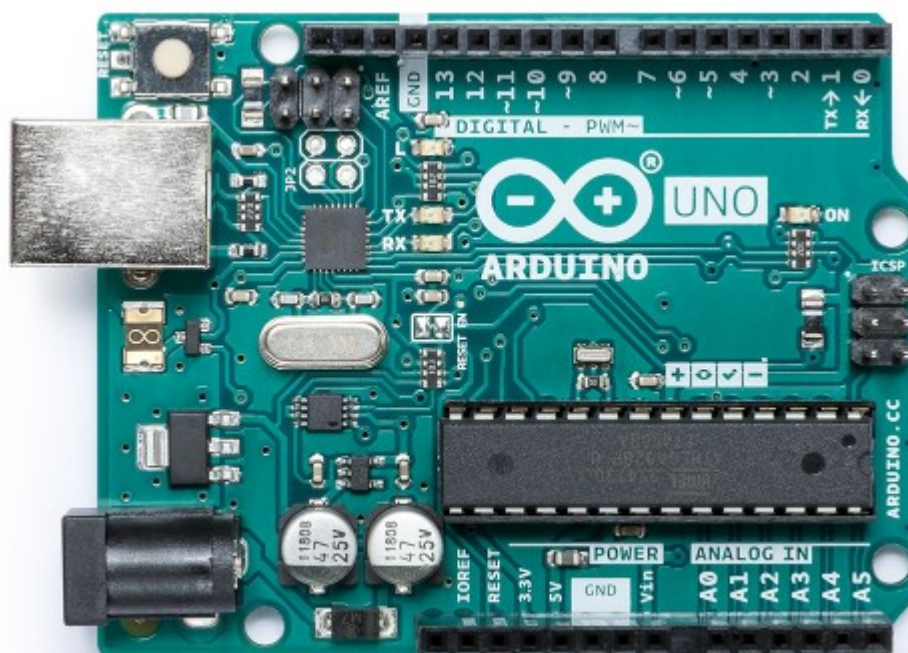


Рис.25 - Arduino Uno (вид сверху)

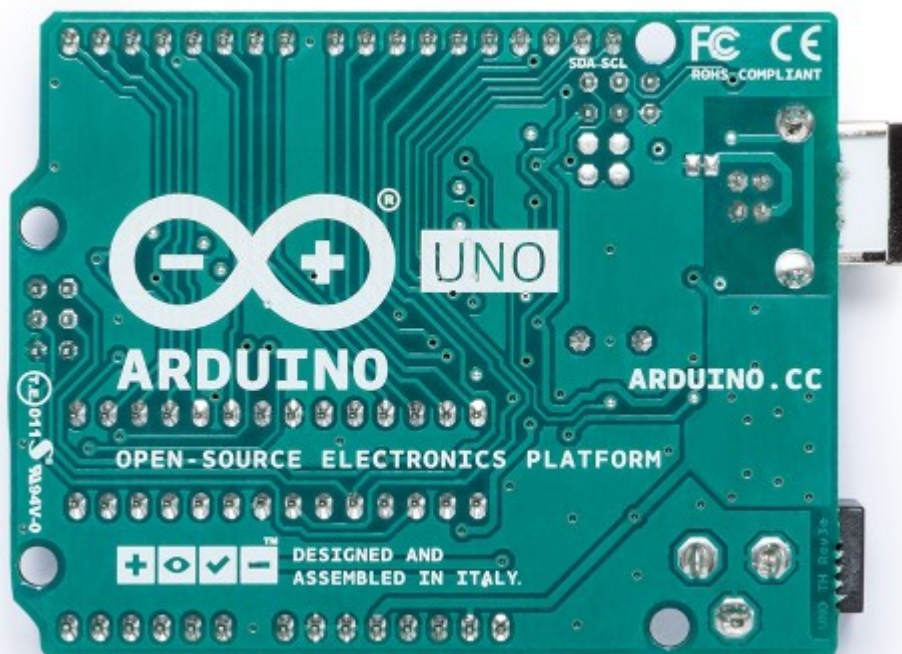


Рис.26 - Arduino Uno (вид снизу)

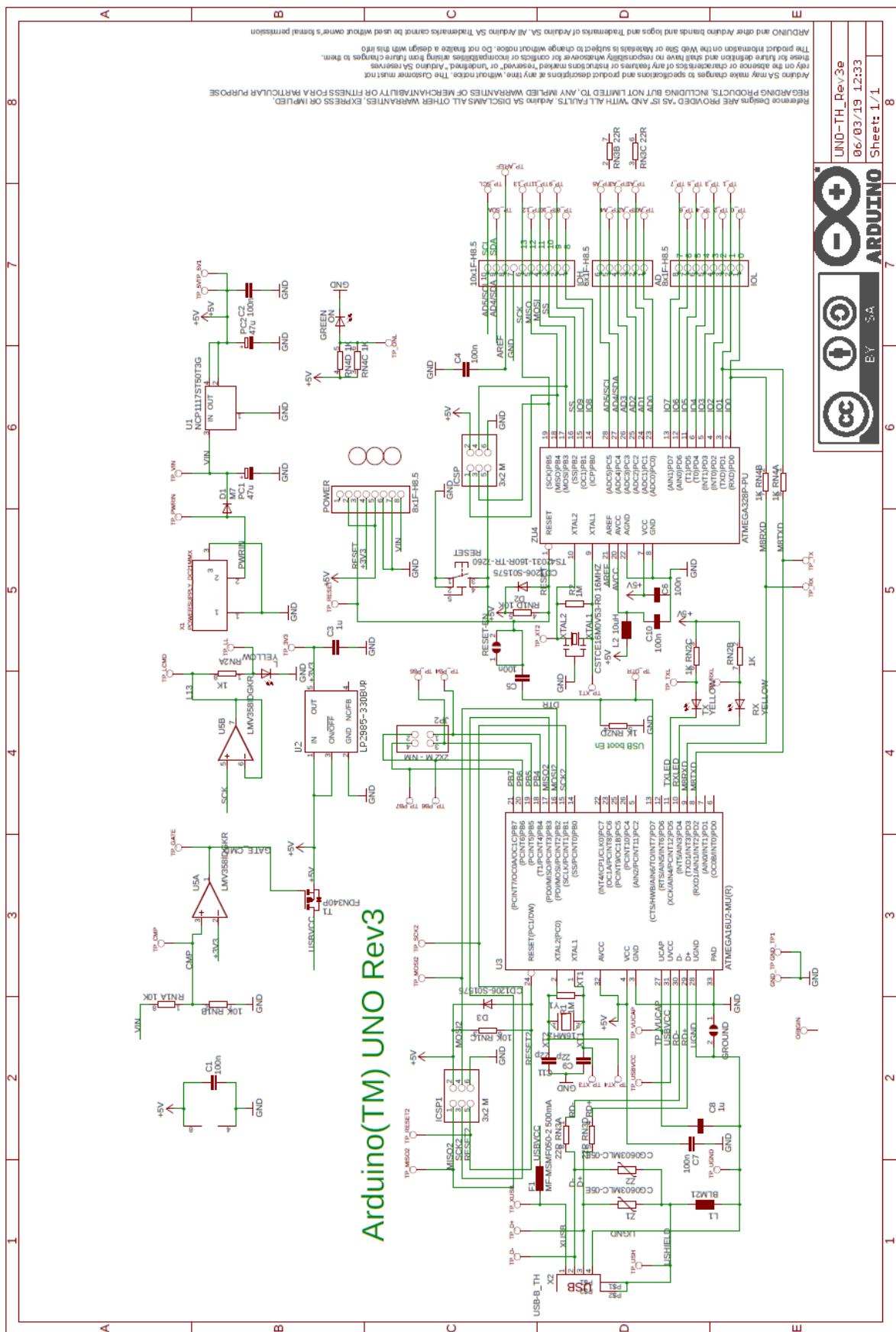


Рис.27 - Принципова схема Arduino UnoR3

Перевагами використання Arduino UnoR3 є відкриті схеми устаткування і специфікації. Arduino Uno виконаний на популярних мікропроцесорах Amtel і ATMEGA. Можна спроектувати на основі наявних схем власний варіант модуля для певних завдань.

Відкритий код програми. Кодування програми може розширюватися на платформі C ++.

Просте і зручне середовище програмування. Оболонка програми є легкою в застосуванні для початківців програмістів, однак має достатню гнучкість для роботи професіоналів. Вона найзручніша для середовища навчання студентів, яким легко буде розібратися в роботі цієї платформи.

Програмування, підключення та живлення виконується одним USB-кабелем, або кабелем, що має адаптер на мікросхемі.

Можливість функціонування на різних видах систем. Програмне забезпечення успішно функціонує на Лінуks, Макінтош, Ява і інших системах, так як має відкритий код. Однак найбільш популярною системою для Arduino стала система Windows.

Arduino Uno має невелику ціну. Ми отримуємо готовий закінчений пристрій, що не вимагає допоміжного обладнання, дорогих програматорів, платних програм.

Розділ 5. Конструкція керованого регулятора температури для дослідної нагрівальної установки

5.1 Функціональна схема конструкції

Конструкція керованого регулятора температури складається з шести основних взаємодіючих між собою компонентів (Рис.28).

- Реле
- Термопара з цифровим підсилювачем
- LCD дисплей;
- I2C модуль;
- Arduino uno;
- Функціональні кнопки (3шт.).

Живлення може відбуватися через USB підключення або від адаптера AC/DC.

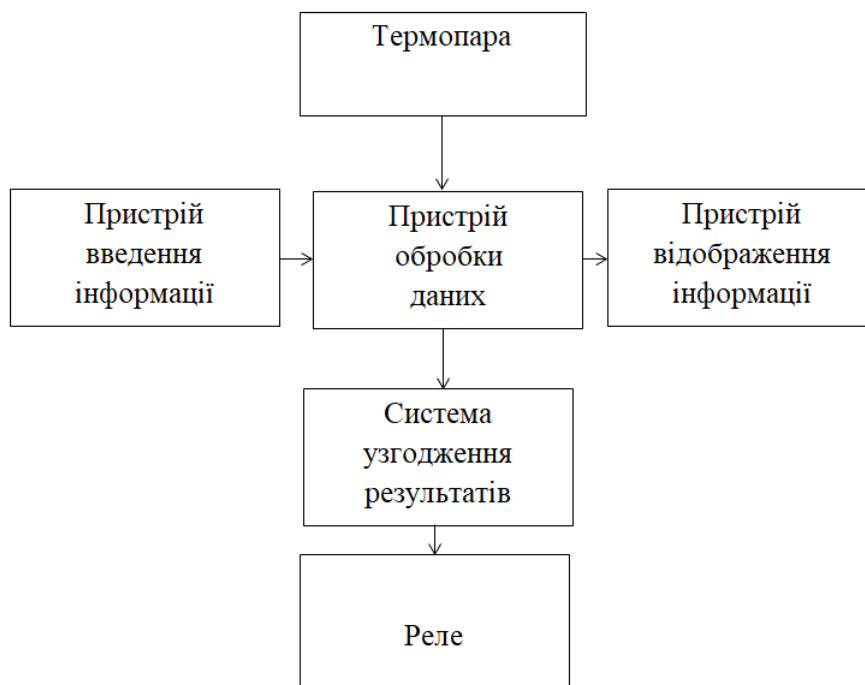


Рис.28 – Структурна схема конструкції регулятора температури

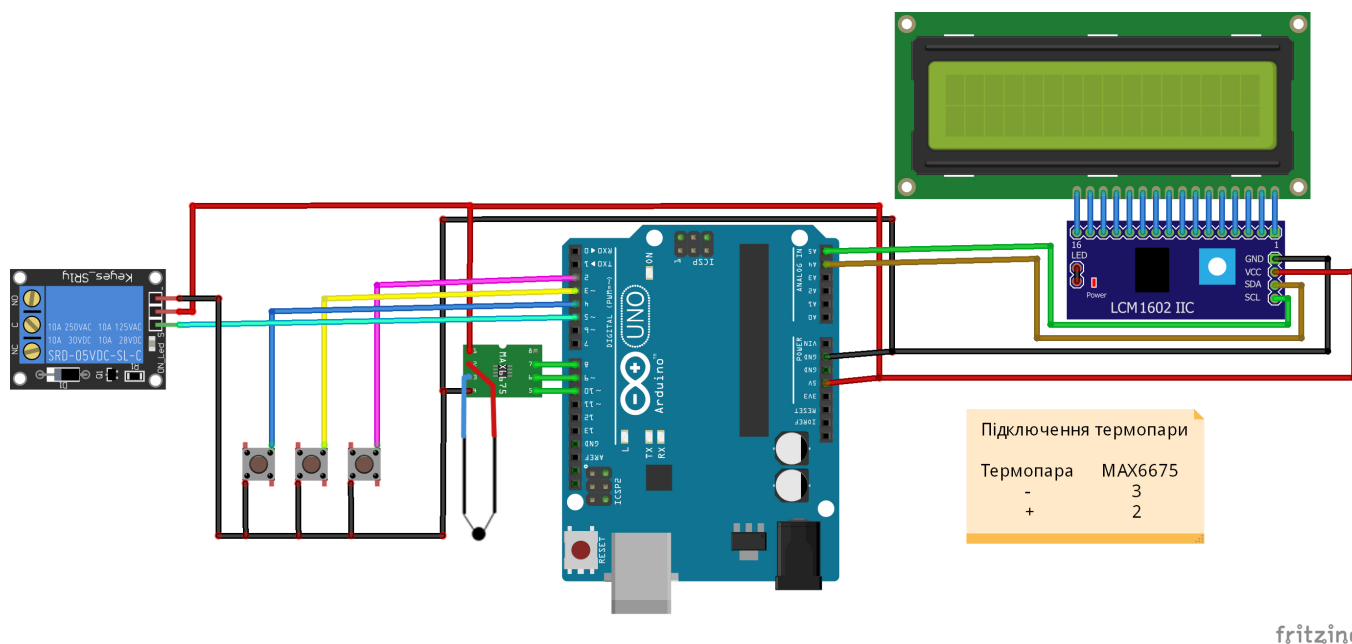


Рис.29 - Візуалізація схеми підключення компонентів
(створено в програмі «fritzing»)

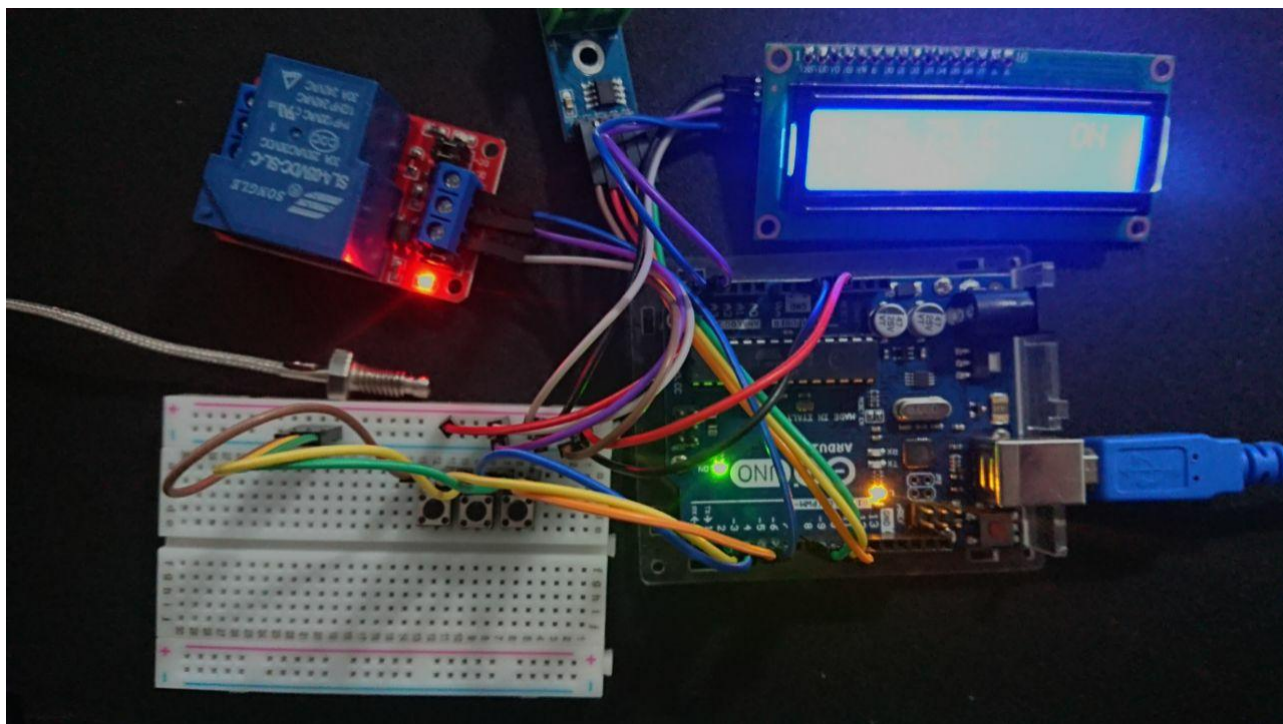


Рис.30 - Фото робочого макету

5.2 Написання коду програми в Arduino IDE

Arduino IDE – програмне середовище розробки, що використовує C++ . З її допомогою програмують всі Arduino-сумісні плати.

Використовуючи Arduino IDE, можна опиратися на попередньо отриманні під час навчання знання з мови програмування C++. Програмне забезпечення є повністю безкоштовним і представлена з відкритим вихідним кодом.

Для нормального функціонування LCD дисплею та термопари з цифровим підсилювачем на MAX6675 необхідно завантажити потрібні бібліотеки з офіційного сайту Arduino. Ці бібліотеки завантажуються для того, щоб у середовищі розробки з'явилися необхідні команди щодо керування вище вказаними модулями. Код програми (Рис.31) знаходиться у додатку 1.

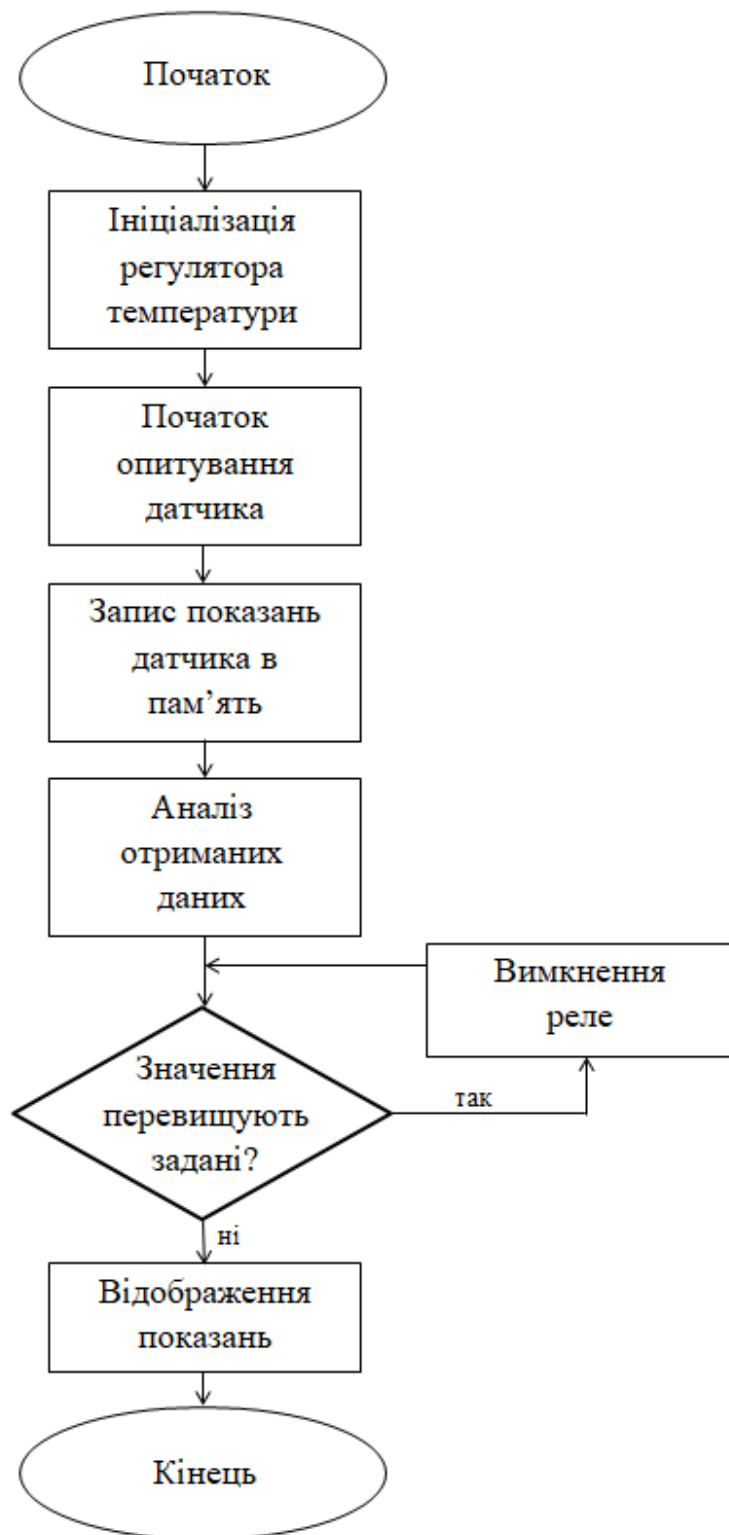


Рис.31 - Блок-схема алгоритму роботи програми

У програмному середовищі розробки було створено головний екран, на якому відображується:

- температура термопарі в реальному часі;
- максимальний температурний поріг , при якому вимкнеться реле;
- стан реле в даний момент («ON» або « OFF»).
- Також створено два пункти функції «MENU»:
 - Histeresis (гістерезис);
 - Temperature (максимально можлива температура до якої відбуватиметься нагрівання, та після досягнення якої вимкнеться реле).

Програмним способом було реалізовано функцію гістерезису з можливістю його точного налаштування, для регулювання циклів вмикання та вимикання нагрівальної установки за допомогою реле. Навігація в меню (Рис.29) відбувається за допомогою двох кнопок: B2 та B3. Підтвердження налаштування – натиск на ліву крайню кнопку B1.

Покази температури термопарі оновлюються кожні 0,5с.

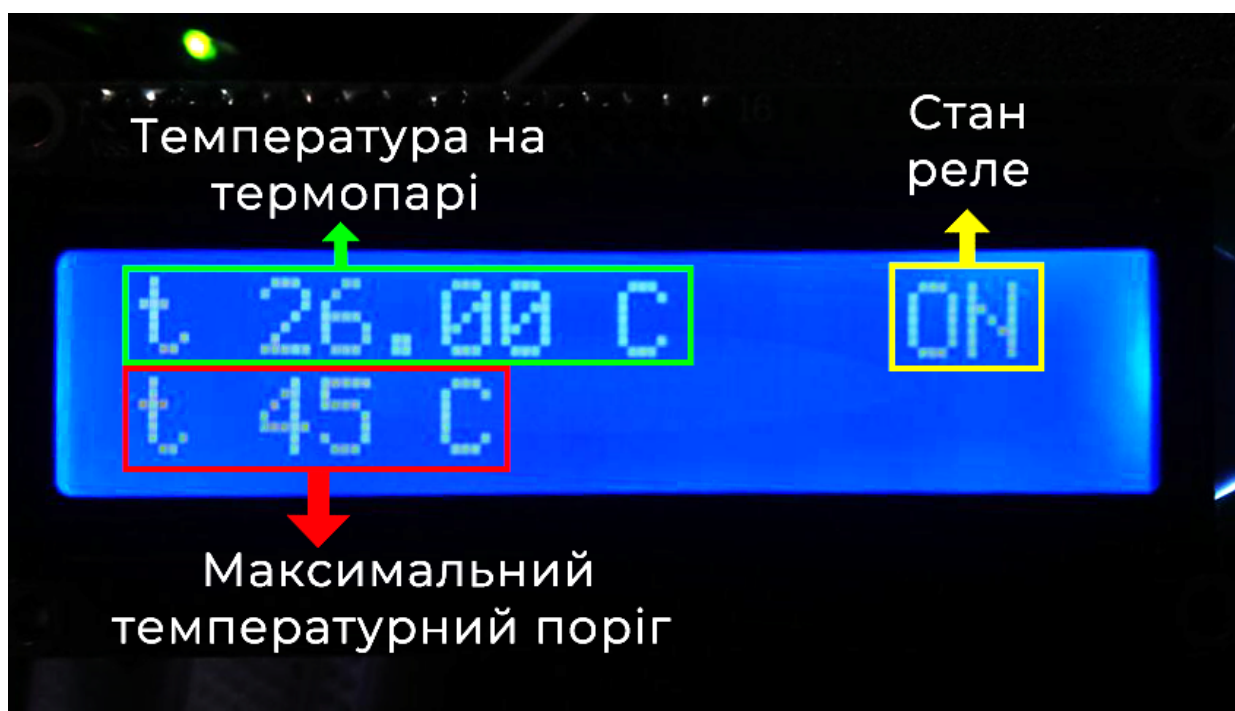


Рис.32 - Головне меню регулятор температури.



Рис.33 - Пункт меню - «Temperature»



Рис.34 - Пункт меню – «Hysteresis»

5.3 Поверхневий монтаж компонентів

Одним із можливих застосувань керованого регулятора температури – регулювання температури при технології пайки SMD компонентів. Для цієї задачі необхідно дотримуватися складного температурного режиму, так званого профілю пайки.

Розглянемо профіль пайки (Рис.35) для паяльної пасти EFD Solderplus 62NCLR-A, з наступним складом : Олово (62%), Свинець (36%), Срібло (2%).

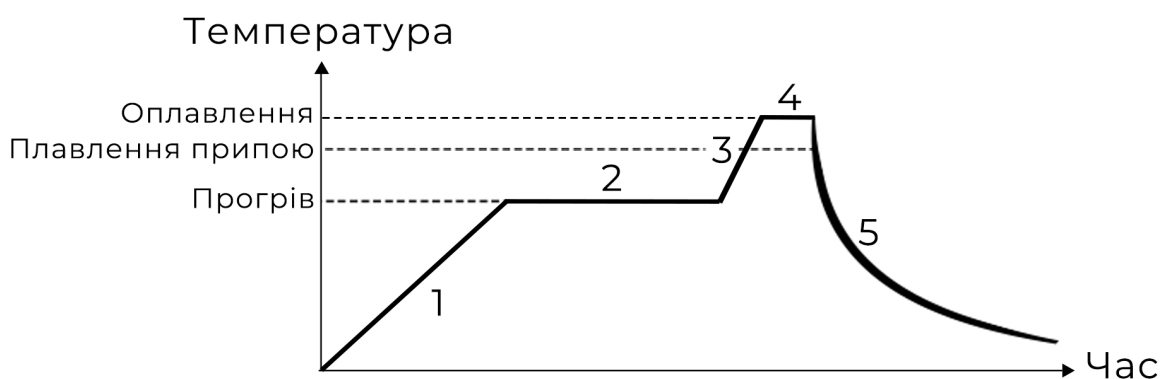


Рис.35 – Профіль пайки SMD компонентів.

Профіль пайки має 5 етапів.

– Розігрів (1). Повільне та обережне підвищення температури. Важливо уникати механічних пошкоджень через нерівномірний нагрів (25 секунд до 100°C);

– Плато (2). Під час цієї фази всі компоненти повинні встигнути прогрітися до температури плавлення припою (150 секунд від 100°C до 150°C);

– Швидкий розігрів (3). Швидкий розігрів та початок лавинообразного оплавлення припою (75 секунд до 200°C);

– Оплавлення (4). Невелика пауза для того, щоб все встигло оплавитися. Перетримка цієї фази може пошкодити компоненти (10 секунд від 200°C до 220°C);

– Охолодження (5). Вільне охолодження (до кімнатної температури).

При груповій пайці компонентів важливо виконувати ряд технологічних вимог. Дискретні компоненти і мікросхеми з кроком виводів більше 1 мм повинні бути розміщені так, щоб виводи компонентів не виходили за межі контактної майданчику (Рис.32). Оптимальне симетричне розташування компонентів. Такі компоненти паяються методом групової пайки в конвекційних печах.

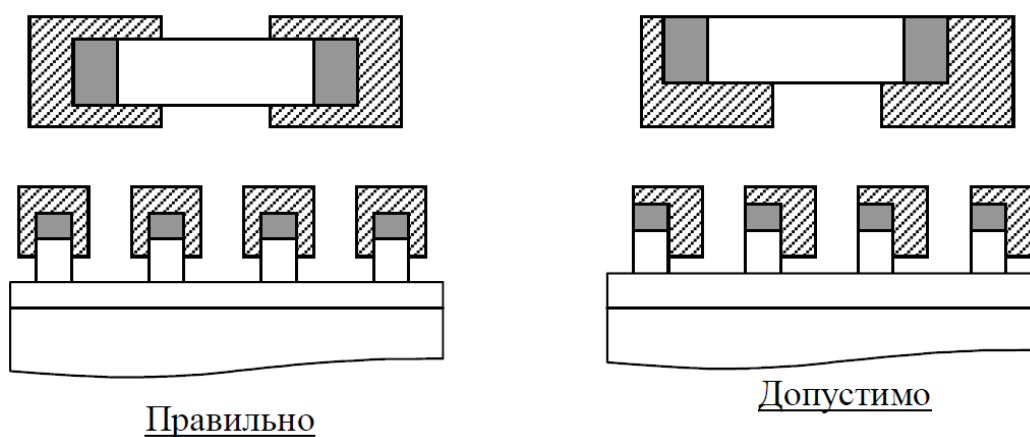


Рис.36 - Допуски розміщення SMD компонентів

Мікросхеми з більш дрібним кроком. Нові мікросхеми процесорів і пам'яті мають, як правило, крок виводів від 0,65мм до 0,8 мм і велика кількість виводів на 2 або 4 сторони. При ручному позиціонуванні для допоміжної фіксації компонентів використовується спеціальна мастика. Виводи паяються за допомогою фломастера з флюсом NC або RMA і наконечником "мікрохвиля" паяльної станції. При пайці груповим методом потрібне застосування тонкого (0,3 мм) наконечника для диспенсера.

У зв'язку з переважним використанням SMD - монтажу звичайні компоненти використовуються виключно через велику розсіювану потужність і в високовольних ланцюгах для забезпечення ізоляційного зазору. В більшості випадків в якості додаткового відводу тепла використовуються широкі

провідники плати, тому при монтажі важливу роль відіграє скорочення довжини виводів компонентів для зменшення теплового опору. Також в більшості випадків слід прагнути до забезпечення мінімальної висоти монтажу. Масивні і великі компоненти (електроліт. конденсатори, трансформатори і т.д.) монтують впритул до плати для підвищення стійкості до ударів і вібрації.

Компоненти в DIP-корпусах необхідно монтувати якомога ближче до плати, щоб їх виводи були якомога коротшими (Рис.33) для забезпечення мінімального теплового опору.

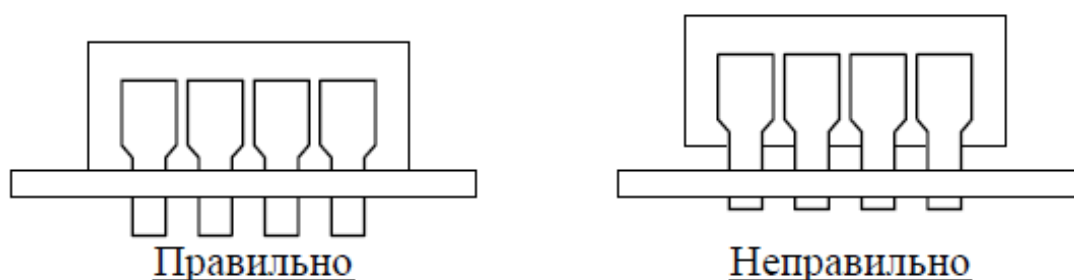


Рис.37 - Монтаж компонентів в DIP-корпусах

Компоненти з аксіальними виводами монтуються, як правило, вертикально для кращого повітряного охолодження. При цьому діоди і стабілітрони повинні монтуватися катодом (смугою) вгору з зазором від 1 до 2 мм між корпусом приладу і платою. При достатній кількості місця для розміщення вони можуть монтуватися горизонтально (Рис.34). В цьому випадку між такими компонентами і платою повинен залишатися повітряний зазор від 1 до 2 мм.

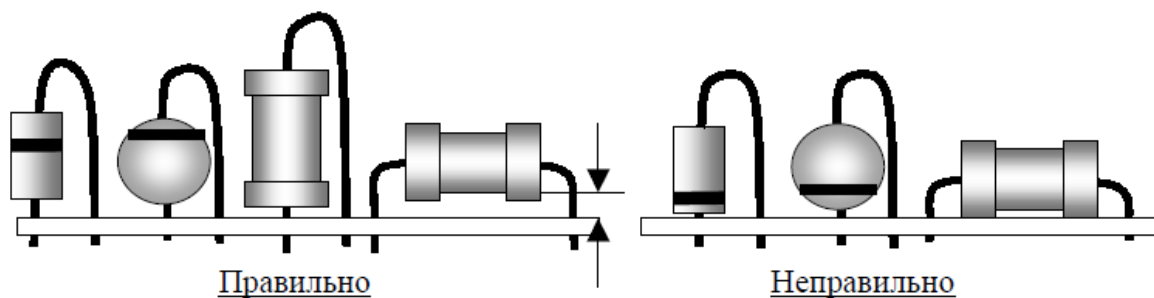


Рис.38 - Монтаж компонентів з аксіальними виводами

Компоненти з виводами в одну сторону - керамічні та плівкові конденсатори, резонатори, варистори, напівпровідникові запобіжники, монтуються з умовою забезпечення мінімального зазору (приблизно 1 мм) між корпусом компонента і платою за відсутності механічної напруги виводів.

Літієві джерела живлення мають дуже велику чутливість до струмів розряду. Плати з встановленими літієвими елементами не можна мити з використанням струмопровідних розчинів, тому їх слід монтувати в останню чергу, після всіх промивок.

Кварц в низьких корпусах Н49U слід встановлювати з мінімальним зазором до друкованої плати. Якщо під кварцом немає перехідних отворів і струмопровідних провідників, то рекомендується установка впритул до плати. Кварц в високих корпусах Н49 і часові кварци слід укласти на друковану плату у вільну сторону або на земляний провідник. По можливості, корпус кварцу слід фіксувати краплею припою до земляного провідника.

Світлодіоди в нових, подовжених корпусах діаметром 5 мм слід монтувати впритул до плати (Рис.35), в мініатюрних 3 мм корпусах впаюються по фіксуючі виступи на виводах. Виводи анодів імпортованих світлодіодів довші.

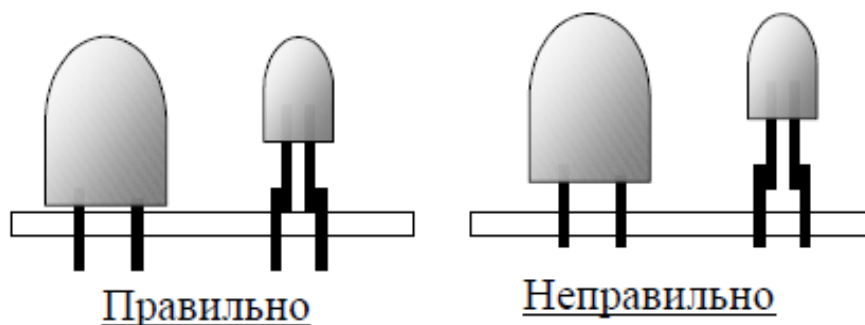


Рис.39 - Монтаж світлодіодів

Електролітичні конденсатори з штировими виводами використовуються в потужних високочастотних перетворювачах і високовольних ланцюгах. Ці конденсатори мають модернізований корпус з газовидільним клапаном і картонній або гумовій настановній прокладці та розраховані на монтаж впритул

до плати (Рис.36). Зазвичай на корпусі маркується негативний вивід конденсатора. Іноді для зниження загальної висоти монтажу їх укладають на плату.

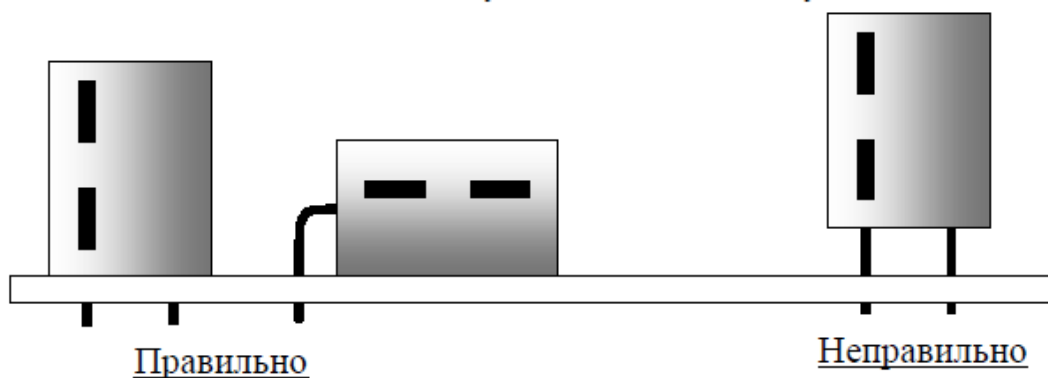


Рис.40 - Монтаж конденсаторів

Механічні компоненти - кнопки, роз'єми, джампери, клемники зазвичай мають механічні прив'язки до кількох елементів конструкції - платам або елементам корпусу. Точність їх розміщення задається конструктором по точках прив'язки на платі. Якщо немає конструкторських вимог по установці даного компонента, то необхідно монтувати компонент по монтажним точкам, без перекосів і зазорів.

Загальні висновки

В дипломній роботі розглянуті види існуючих терморегуляторів та датчиків температури, і висвітлені принципи їх роботи.

Для датчика температури запропонована термопара К-типу з цифровим підсилювачем, як найефективніша для застосування в діапазоні температур від 0 до 600°C.

Обрано мікроконтролерний управляючий компонент для регулятора температури на основі платформи Arduino Uno R3.

Написано програмний код, за допомогою якого реалізовано управління нагрівом в залежності від показань термопари.

Організовано відображення інформації на LCD дисплеї, забезпечена навігація по декількох функціях в меню.

Мета роботи була у цілому досягнена. Розроблений керований регулятор температури для дослідної нагрівальної установки відповідає вимогам завдання на дипломне проектування.

Основними перевагами розробленого регулятора є простота реалізації схеми, мала вартість, достатня точність установки температури та можливість регулювати температуру у широкому діапазоні.

Дану конструкцію регулятора температури можна використовувати для лабораторних досліджень: наприклад, для дослідження теплофізичних властивостей матеріалів та компонентів, для теплових випробувань розроблюваних електронних пристроїв та систем, для підтримання заданих температурних режимів електричних нагрівачів технологічного призначення.

Перелік джерел та посилань

1. Температурные измерения: Справочник / О. А. Геращенко, А. Н. Гордов, Б. И. Стадник, В. И. Лах, Я. Т. Луцик, Н. А. Ярышев, А. К. Еремина. — К.: Наук. думка, 1989. — 494 с.

2. «What is Pyrometer : Working Principle and Its Types».
URL: <https://www.elprocus.com/what-is-pyrometer-working-principle-and-its-types/>
(дата доступу 18.05.2020).

3. Application of Thermo-Fluidic Measurement Techniques .
URL: <https://www.sciencedirect.com/science/article/pii/B978012809731100006X>
(дата доступу 25.05.2020).

4. Высокоточный терморегулятор для термостата на ШИМ-контроллере.
URL: <http://radioamator.ru/avtomatika/termoregulatory-termostaty/1815-vysokotochnyj-termoregulyator-dlya-termostata-na-shim-kontrollere>

5. Плата Arduino UnoR3: схема, описание, подключение устройств.
URL: <https://arduinomaster.ru/platy-arduino/plata-arduino-uno/>

Додаток 1. Код програми

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "max6675.h"
#include <EEPROM.h>

LiquidCrystal_I2C _lcd1(0x27, 16, 2);
int _dispTempLength1=0;
boolean _isNeedClearDisp1;

MAX6675 MAX6675_9_10_8(10, 9, 8);

unsigned long MAX6675_StTime;
bool MAX6675_CanPoll = 0;
const byte _menuParametrsArray[] PROGMEM = {1, 3, 2, 0, 0, 0, 3, 4, 1, 4, 2, 0, 2, 3,
1, 0, 0, 0, 2, 4, 1, 4, 1, 0};
int _menuValueArray_int[2];
const int _menuConstantValuesArray_int[] PROGMEM = {1, 350, 50, 0};
const char _flprogMenuNameString1[] PROGMEM = "Temperature";
const char _flprogMenuNameString2[] PROGMEM = "Hysteresis";
const char* const _flprogMenuStringsArray[] PROGMEM = {
_flprogMenuNameString1, _flprogMenuNameString2};
struct _menuItemStricture {
int startInArrayIndex;
};
struct _menuMainStricture {
byte startIndex;
byte stopIndex;
bool isSwitchMenuAroundRing;

```

```

_menuItemStricture currentItem;
};
_menuItemStricture _MenuItems[2];
_menuMainStricture _MainMenus[1];
float _gtv4;
bool _gtv5;
bool _gtv7;
bool _bounseInputD2S = 0;
bool _bounseInputD2O = 0;
unsigned long _bounseInputD2P = 0UL;
bool _bounseInputD3S = 0;
bool _bounseInputD3O = 0;
unsigned long _bounseInputD3P = 0UL;
int _disp6oldLength = 0;
int _disp1oldLength = 0;
int _disp13oldLength = 0;
bool _trgs2 = 0;
int _disp10oldLength = 0;
int _disp8oldLength = 0;
bool _tim2I = 0;
bool _tim2O = 0;
unsigned long _tim2P = 0UL;
bool _trgrt1 = 0;
bool _trgrt1I = 0;
bool _bounseInputD4S = 0;
bool _bounseInputD4O = 0;
unsigned long _bounseInputD4P = 0UL;
bool _MenuBlock_255966044_AMO_129592179 = 0;
bool _MenuBlock_255966044_AMO_103176861 = 0;
String _MenuBlock_255966044_MNO;

```

```

String _MenuBlock_255966044_VNO;
bool _MenuBlock_255966044_OEIS = 0;
bool _MenuBlock_255966044_OMUIS = 0;
bool _MenuBlock_255966044_OMDIS = 0;
bool _MenuBlock_255966044_OVUIS = 0;
bool _MenuBlock_255966044_OVDIS = 0;
unsigned long _MenuBlock_255966044_VUDST = 0UL;
unsigned long _MenuBlock_255966044_VUDSPT = 0UL;
byte _MenuBlock_255966044_VUDSM = 0;
int _MenuBlock_255966044_VUDPT = 0;
int _disp12oldLength = 0;
unsigned long MAX6675_9_10_8_TT = 0UL;
float MAX6675_9_10_8_Output = 0.00;
int _disp11oldLength = 0;
bool _trgt2 = 0;
bool _trgt2I = 0;
int _disp2oldLength = 0;
int _disp3oldLength = 0;

void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
  pinMode(5, OUTPUT);
  digitalWrite(5, 0);

  Wire.begin();
  delay(10);

```

```

MAX6675_StTime = millis();
MAX6675_9_10_8_TT = millis() +500;
_lcd1.init();
_lcd1.backlight();
_bounceInputD2O = digitalRead(2);
_bounceInputD3O = digitalRead(3);
_bounceInputD4O = digitalRead(4);
_MenuItems[0].startInArrayIndex = 0;
_MenuItems[1].startInArrayIndex = 12;
_MainMenus[0].startIndex = 1;
_MainMenus[0].isSwitchMenuAroundRing = 1;
_MainMenus[0].stopIndex = 2;
_MainMenus[0].currentItem = _MenuItems[0];
_menuValueArray_int[0] = (readIntegerFromEEPROM(0, 0, 0x0));
_menuValueArray_int[1] = (readIntegerFromEEPROM(2, 0, 0x0));

}

void loop()
{

bool _tempVariable_bool;
byte _tempVariable_byte;
if (! MAX6675_CanPoll) { MAX6675_CanPoll = _isTimer(MAX6675_StTime, 500);}
if (_isNeedClearDisp1) {_lcd1.clear(); _isNeedClearDisp1= 0;}

bool _bounceInputTmpD2 = (digitalRead (2));

if (_bounceInputD2S)
{
if (millis() >= (_bounceInputD2P + 40))

```

```

{ _bounceInputD2O = _bounceInputTmpD2; _bounceInputD2S = 0; }
}
else
{
if ( _bounceInputTmpD2 != _bounceInputD2O )
{ _bounceInputD2S = 1; _bounceInputD2P = millis(); }
}
bool _bounceInputTmpD3 = (digitalRead (3));

if ( _bounceInputD3S )
{
if ( millis() >= ( _bounceInputD3P + 40 ) )
{ _bounceInputD3O = _bounceInputTmpD3; _bounceInputD3S = 0; }
}
else
{
if ( _bounceInputTmpD3 != _bounceInputD3O )
{ _bounceInputD3S = 1; _bounceInputD3P = millis(); }
}
bool _bounceInputTmpD4 = (digitalRead (4));

if ( _bounceInputD4S )
{
if ( millis() >= ( _bounceInputD4P + 40 ) )
{ _bounceInputD4O = _bounceInputTmpD4; _bounceInputD4S = 0; }
}
else
{
if ( _bounceInputTmpD4 != _bounceInputD4O )
{ _bounceInputD4S = 1; _bounceInputD4P = millis(); }
}

```

```
}
```

```
//Плата:1
```

```
//Меню
```

```
if (!(_bounseInputD2O))
```

```
{ if (_tim2I) { if (_isTimer(_tim2P, 2000)) {_tim2O = 1;}} else {_tim2I =1; _tim2P =
millis();}} else {_tim2O = 0; _tim2I = 0;}
```

```
if (_tim2O) { if (_trgrt1I) {_trgrt1 = 0;} else {_trgrt1 = 1; _trgrt1I = 1;} } else {_trgrt1
= 0; _trgrt1I = 0;};
```

```
bool _tmp2 = _trgrt1;
```

```
if (_tmp2) { if (! _trgt2I) _trgt2 = ! _trgt2; }
```

```
_trgt2I = _tmp2;
```

```
_gtv5 = _trgt2;
```

```
if (_gtv5) {
```

```
_tempVariable_bool = 1;
```

```
if ( ! _MenuBlock_255966044_OEIS ) {_MenuBlock_255966044_OEIS = 1;
```

```
}
```

```
_tempVariable_byte =
```

```
pgm_read_byte(&_menuParamtrsArray[((_MainMenus[0].currentItem
).startInArrayIndex)+10]);
```

```
_MenuBlock_255966044_MNO = _readStringFromProgmem
```

```
((char*)pgm_read_word(&(_flprogMenuStringsArray[_tempVariable_byte - 1])));
```

```
_MenuBlock_255966044_VNO = _menuOutputValueString (0);
```

```
_MenuBlock_255966044_AMO_129592179 =
```

```
pgm_read_byte(&_menuParamtrsArray[( _MainMenus[0].currentItem
).startInArrayIndex]) == 1;
```



```

_MenuBlock_255966044_AMO_103176861 =
pgm_read_byte(&_menuParametersArray[( _MainMenus[0].currentItem
).startInArrayIndex]) == 2;
} else {
_tempVariable_bool = 0;
if ( _MenuBlock_255966044_OEIS){
_MenuBlock_255966044_OEIS = 0;
_menuUpdateToEEPROMItems();
}
_MenuBlock_255966044_AMO_129592179 = 0;
_MenuBlock_255966044_AMO_103176861 = 0;
_MenuBlock_255966044_MNO = "";
_MenuBlock_255966044_VNO = "";
}
if(( ( _gtv5) && !( _bounseInputD2O)) ) { if ( ! _MenuBlock_255966044_OMUIS)
{ _MenuBlock_255966044_OMUIS = 1; if ( _tempVariable_bool){ _menuUpEvents(0);
_menuUpdateToEEPROMItems();
}}} else { _MenuBlock_255966044_OMUIS = 0;}
if(0) { if ( ! _MenuBlock_255966044_OMDIS) { _MenuBlock_255966044_OMDIS =
1; if ( _tempVariable_bool){ _menuDownEvents(0);
_menuUpdateToEEPROMItems();
}}} else { _MenuBlock_255966044_OMDIS = 0;}
if(!( _bounseInputD3O)) { if ( ! _MenuBlock_255966044_OVUIS)
{ _MenuBlock_255966044_OVUIS = 1; if ( _tempVariable_bool){ _valueUpEvents(0);
_MenuBlock_255966044_VUDST = millis();} } if ( _tempVariable_bool){
if ( _MenuBlock_255966044_VUDSM == 0) { if
( _isTimer( _MenuBlock_255966044_VUDST , 2000 ))
{ _MenuBlock_255966044_VUDSM = 1; _MenuBlock_255966044_VUDSPT =
millis(); _MenuBlock_255966044_VUDPT = 500;}} } else { if
( _MenuBlock_255966044_VUDSM == 1) { if

```

```

(_isTimer(_MenuBlock_255966044_VUDST , 4000 ))
{ _MenuBlock_255966044_VUDSM = 2; _MenuBlock_255966044_VUDPT = 250; } }
if (_isTimer(_MenuBlock_255966044_VUDSPT , _MenuBlock_255966044_VUDPT
)) { _valueUpEvents(0); _MenuBlock_255966044_VUDSPT = millis();} }
}} else { _MenuBlock_255966044_OVUIS = 0; if( ! (
!(_bounseInputD4O))) { _MenuBlock_255966044_VUDSM = 0; }}
if(!(_bounseInputD4O)) { if ( ! _MenuBlock_255966044_OVDIS)
{ _MenuBlock_255966044_OVDIS = 1; if (_tempVariable_bool){
_valueDownEvents(0); _MenuBlock_255966044_VUDST = millis();}} if
(_tempVariable_bool){
if (_MenuBlock_255966044_VUDSM == 0) { if
(_isTimer(_MenuBlock_255966044_VUDST , 2000 ))
{ _MenuBlock_255966044_VUDSM = 1; _MenuBlock_255966044_VUDSPT =
millis(); _MenuBlock_255966044_VUDPT = 500;}} else { if
(_MenuBlock_255966044_VUDSM == 1) { if
(_isTimer(_MenuBlock_255966044_VUDST , 4000 ))
{ _MenuBlock_255966044_VUDSM = 2; _MenuBlock_255966044_VUDPT = 250; } }
if (_isTimer(_MenuBlock_255966044_VUDSPT , _MenuBlock_255966044_VUDPT
)) { _valueDownEvents(0); _MenuBlock_255966044_VUDSPT = millis();} }
}} else { _MenuBlock_255966044_OVDIS = 0; if( ! (
!(_bounseInputD3O))) { _MenuBlock_255966044_VUDSM = 0; }}
if (_MenuBlock_255966044_AMO_103176861) {
_dispTempLength1 = (String("Temperature")).length();
if (_disp6oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
_disp6oldLength = _dispTempLength1;
_lcd1.setCursor(int((16 - _dispTempLength1)/2), 0);
_lcd1.print(String("Temperature"));
} else {
if (_disp6oldLength > 0) {_isNeedClearDisp1 = 1; _disp6oldLength = 0;}
}
}

```

```

if (_MenuBlock_255966044_AMO_103176861) {
    _dispTempLength1 = (((String("Ust t ")) + ((String((_menuValueArray_int[0]),
    DEC))) + (String(" C")))).length();
    if (_disp1oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
    _disp1oldLength = _dispTempLength1;
    _lcd1.setCursor(0, 1);
    _lcd1.print((((String("Ust t ")) + ((String((_menuValueArray_int[0]), DEC))) +
    (String(" C")))));
} else {
    if (_disp1oldLength > 0) {_isNeedClearDisp1 = 1; _disp1oldLength = 0;}
}

if (_MenuBlock_255966044_AMO_129592179) {
    _dispTempLength1 = (String("Hysteresis")).length();
    if (_disp2oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
    _disp2oldLength = _dispTempLength1;
    _lcd1.setCursor(int((16 - _dispTempLength1)/2), 0);
    _lcd1.print(String("Hysteresis"));
} else {
    if (_disp2oldLength > 0) {_isNeedClearDisp1 = 1; _disp2oldLength = 0;}
}

if (_MenuBlock_255966044_AMO_129592179) {
    _dispTempLength1 = (((String("Ust ") + ((String((_menuValueArray_int[1]), DEC)))
    + (String(" C")))).length();
    if (_disp3oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
    _disp3oldLength = _dispTempLength1;
    _lcd1.setCursor(0, 1);
    _lcd1.print((((String("Ust ") + ((String((_menuValueArray_int[1]), DEC))) + (String("
    C")))));
} else {
    if (_disp3oldLength > 0) {_isNeedClearDisp1 = 1; _disp3oldLength = 0;}
}

```

```

}
if (1) {
    _dispTempLength1 = ((_MenuBlock_255966044_VNO)).length();
    if (_disp8oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
    _disp8oldLength = _dispTempLength1;
    _lcd1.setCursor(12, 1);
    _lcd1.print((_MenuBlock_255966044_VNO));
} else {
    if (_disp8oldLength > 0) {_isNeedClearDisp1 = 1; _disp8oldLength = 0;}
}

//Плата:2
//Датчик
if (!(_gtv5)) {
    _dispTempLength1 = (((String("t ")) + ((String((_menuValueArray_int[0]), DEC))) +
    (String(" C")))).length();
    if (_disp11oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
    _disp11oldLength = _dispTempLength1;
    _lcd1.setCursor(0, 1);
    _lcd1.print((((String("t ")) + ((String((_menuValueArray_int[0]), DEC))) + (String("
    C")))))));
} else {
    if (_disp11oldLength > 0) {_isNeedClearDisp1 = 1; _disp11oldLength = 0;}
}

if (!(_gtv5)) {
    _dispTempLength1 = (((String("t ")) + ((_floatToStringWitRaz(_gtv4,2))) + (String("
    C")))).length();
    if (_disp10oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
    _disp10oldLength = _dispTempLength1;
    _lcd1.setCursor(0, 0);

```

```

_lcd1.print((((String("t ")) + (( _floatToStringWitRaz(_gtv4,2))) + (String(" C")))));
} else {
if (_disp10oldLength > 0) {_isNeedClearDisp1 = 1; _disp10oldLength = 0;}
}
if (1) {
if ( _isTimer(MAX6675_9_10_8_TT, 500 )) {
MAX6675_9_10_8_TT = millis();
if(MAX6675_CanPoll) {MAX6675_9_10_8_Output =
MAX6675_9_10_8.readCelsius();}
}
}
_gtv4 = MAX6675_9_10_8_Output;

//Плата:3
//Виконуюча частина
if (( !(_gtv5)) && (_gtv7) )) {
_dispTempLength1 = (String("ON")).length();
if (_disp12oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
_disp12oldLength = _dispTempLength1;
_lcd1.setCursor(13, 0);
_lcd1.print(String("ON"));
} else {
if (_disp12oldLength > 0) {_isNeedClearDisp1 = 1; _disp12oldLength = 0;}
}
if (( !(_gtv5)) && !( _gtv7) )) {
_dispTempLength1 = (String("OFF")).length();
if (_disp13oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}
_disp13oldLength = _dispTempLength1;
_lcd1.setCursor(13, 0);
_lcd1.print(String("OFF"));
}

```

```

} else {
if (_disp13oldLength > 0) {_isNeedClearDisp1 = 1; _disp13oldLength = 0;}
}
if((_gtv4) < (((_menuValueArray_int[0]))-((_menuValueArray_int[1])))) _trgs2 = 1;
if((_gtv4) > ((_menuValueArray_int[0]))) _trgs2 = 0;
_gtv7 = _trgs2;
digitalWrite(5, _gtv7);

```

```

}

```

```

String _floatToStringWitRaz(float value, int raz)

```

```

{

```

```

return String(value,raz);

```

```

}

```

```

bool _isTimer(unsigned long startTime, unsigned long period )

```

```

{

```

```

unsigned long currentTime;

```

```

currentTime = millis();

```

```

if (currentTime>= startTime) {return (currentTime>=(startTime + period));} else

```

```

{return (currentTime >=(4294967295-startTime+period));}

```

```

}

```

```

String _readStringFromProgmam (char *string)

```

```

{

```

```

String result = String("");

```

```

while (pgm_read_byte(string)!='\0')
{
result=result+ char(pgm_read_byte(string));
string++;
}
return result;
}

void _menuUpEvents (byte menuIndex)
{
byte tempIndex =
pgm_read_byte(&_menuParamsArray[((_MainMenus[menuIndex]).currentItem).start
InArrayIndex]);
byte parIndex = pgm_read_byte(&_menuParamsArray[
((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+5]);
byte parentStartIndex;
byte parentStopIndex;
if (parIndex == 0) {parentStartIndex = ( _MainMenus[menuIndex]).startIndex;
parentStopIndex = ( _MainMenus[menuIndex]).stopIndex;}
else { parentStartIndex = pgm_read_byte(&_menuParamsArray[(
_ MenuItems[parIndex - 1]).startInArrayIndex)+3]);
parentStopIndex = pgm_read_byte(&_menuParamsArray[( _ MenuItems[parIndex-
1]).startInArrayIndex)+4]); }
if (tempIndex == parentStartIndex)
{if((_MainMenus[menuIndex]).isSwitchMenuAroundRing)
{(_MainMenus[menuIndex]).currentItem = _ MenuItems[parentStopIndex -1]; return;}
else{return;}}
(_MainMenus[menuIndex]).currentItem = _ MenuItems[tempIndex - 2];
return;
}

void _menuDownEvents (byte menuIndex)

```

```

{
byte tempIndex =
pgm_read_byte(&_menuParamsArray[((_MainMenus[menuIndex]).currentItem).start
InArrayIndex]);
byte parIndex =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+5]);
byte parentStartIndex;
byte parentStopIndex;
if (parIndex == 0) {parentStartIndex = (_MainMenus[menuIndex]).startIndex;
parentStopIndex = (_MainMenus[menuIndex]).stopIndex;}
else { parentStartIndex = pgm_read_byte(&_menuParamsArray[(
_ MenuItems[parIndex-1]).startInArrayIndex)+3]);
parentStopIndex = pgm_read_byte(&_menuParamsArray[(_ MenuItems[parIndex-
1]).startInArrayIndex)+4]); }
if (tempIndex == parentStopIndex)
{if((_MainMenus[menuIndex]).isSwitchMenuAroundRing)
{(_MainMenus[menuIndex]).currentItem = _MenuItems[parentStartIndex -1]; return;}
else{return;}}
(_MainMenus[menuIndex]).currentItem = _MenuItems[tempIndex];
return;
}

void _valueUpEvents (byte menuIndex)
{
byte valIndex =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+2]);
byte itemType =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+1]);

```



```

byte indexMax =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+6]);
byte indexStep =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+8]);
if (itemType == 3) {if (! indexMax == 0) { if ( !
(int(pgm_read_word(&_menuConstantValuesArray_int[indexMax - 1])) >
int(_menuValueArray_int[valIndex - 1]))) {return;} }
_menuValueArray_int[valIndex - 1] = _menuValueArray_int[valIndex - 1] +
(pgm_read_word(&_menuConstantValuesArray_int[indexStep - 1])); }
}
void _valueDownEvents (byte menuIndex)
{

byte valIndex =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+2]);
byte itemType =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+1]);
byte indexMin =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+7]);
byte indexStep =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+8]);
if (itemType == 3) { if (! indexMin == 0) { if ( !
((int(pgm_read_word(&_menuConstantValuesArray_int[indexMin - 1]))) <
int(_menuValueArray_int[valIndex - 1]))) {return;} }

```

```

_menuValueArray_int[valIndex - 1] = _menuValueArray_int[valIndex - 1] -
(pgm_read_word(&_menuConstantValuesArray_int[indexStep - 1])); }
}

String _menuOutputValueString (byte menuIndex)
{
byte itemType = pgm_read_byte(&_menuParamsArray[
((( _MainMenus[menuIndex]).currentItem).startInArrayIndex)+1]);
byte valIndex =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+2]);
byte indexMin =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+7]);
byte indexMax =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+6]);
if( valIndex == 0){ return "";}
byte convFormat =
pgm_read_byte(&_menuParamsArray[((( _MainMenus[menuIndex]).currentItem).sta
rtInArrayIndex)+9]);
if(itemType == 3) { return _convertNamber(itemType, convFormat, valIndex,
indexMax, indexMin);}
return "";
}

void _menuUpdateToEepromItems()
{
(updateIntegerToEEPROM(0, 0, 0x0, ((_menuValueArray_int[0]))));
(updateIntegerToEEPROM(2, 0, 0x0, ((_menuValueArray_int[1]))));
}

```

```

String _convertNamber(byte itemType, byte convFormat, byte valIndex, byte
indexMax, byte indexMin)
{
if (itemType== 3) {
if (convFormat == 4) { return String((_menuValueArray_int[valIndex - 1 ]),DEC);}
if (convFormat == 5) { return String((_menuValueArray_int[valIndex - 1]),HEX);}
if (convFormat == 6) { return String((_menuValueArray_int[valIndex -1]),BIN);}
}
}

byte readByteFromEEPROM(int adres, byte bitAdres, byte chipAdres)
{
return EEPROM.read(adres);
}

void updateByteToEEPROM(int adres, byte bitAdres, byte chipAdres, byte value)
{
return EEPROM.update(adres, value);
}

int readIntegerFromEEPROM(int adres, byte bitAdres, byte chipAdres)
{
byte hight = readByteFromEEPROM( adres, bitAdres, chipAdres);
byte low = readByteFromEEPROM( (adres+1), bitAdres, chipAdres);
return word(hight, low);
}

void updateIntegerToEEPROM(int adres, byte bitAdres, byte chipAdres, int value)
{
updateByteToEEPROM( adres, bitAdres, chipAdres, highByte(value));
updateByteToEEPROM( (adres+1), bitAdres, chipAdres, lowByte(value));
}

```